

z/OS
3.2

*Security Server RACF Macros and
Interfaces*



Note

Before using this information and the product it supports, read the information in [“Notices” on page 683](#).

This edition applies to IBM® z/OS® 3.2 (5655-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2025-09-30

© **Copyright International Business Machines Corporation 1994, 2025.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures.....	ix
Tables.....	xi
About this document.....	xxv
Purpose of this document.....	xxv
Who should use this document.....	xxv
How to use this document.....	xxv
z/OS information.....	xxvi
Summary of changes.....	xxvii
Summary of changes for z/OS 3.2.....	xxvii
Summary of changes for z/OS 3.1.....	xxviii
Chapter 1. RACF customization macros.....	1
ICHERCDE macro.....	1
ICHNCONV macro.....	10
ICHNCONV coding guideline.....	10
ICHNCONV DEFINE.....	10
ICHNCONV SELECT.....	11
ICHNCONV ACTION.....	17
ICHNCONV END.....	18
ICHNCONV FINAL.....	18
Example of a naming convention table.....	18
ICHRFRTB macro.....	19
Chapter 2. Panel driver interface module (ICHSPF03).....	21
Invoking the panel driver interface.....	21
Panel mapping table.....	21
The ISPLINK call.....	22
Example of a RACF panel interface coding sequence.....	23
Chapter 3. Profile name list service routine (IRRPNL00).....	25
Invoking the profile name list service routine.....	25
Format of returned profile name list.....	26
Return codes.....	26
Chapter 4. Date conversion routine.....	29
Invoking the date conversion routine.....	29
Format of returned converted date.....	29
Return code.....	29
Chapter 5. SMF records.....	31
Record type 80: RACF processing record.....	31
Format of SMF type 80 records.....	33
Table of event codes and event code qualifiers.....	41
Table of relocate section variable data.....	56
Table of extended-length relocate section variable data.....	65

Table of data type 6 command-related data.....	81
Record type 81: RACF initialization record.....	136
Record type 83: Security events.....	145
Product section.....	147
Security section.....	147
Relocate sections.....	154
Reformatted RACF SMF records.....	156
Reformatted process records.....	156
Reformatted status records.....	161

Chapter 6. The format of the unloaded SMF type 80 data..... 169

IRRADU00 record format.....	169
XML grammar.....	170
Steps for converting RACF field names to XML tag names.....	170
The format of the header portion of the unloaded SMF type 30 and type 80.....	171
Event codes.....	173
Record extensions.....	177
The JOBINIT record extension.....	177
The ACCESS record extension.....	182
The ADDVOL record extension.....	185
The RENAMEDS record extension.....	187
The DELRES record extension.....	189
The DELVOL record extension.....	191
The DEFINE record extension.....	192
The ADDSD record extension.....	194
The ADDGROUP record extension.....	195
The ADDUSER record extension.....	197
The ALTDSD record extension.....	199
The ALTGROUP record extension.....	200
The ALTUSER record extension.....	202
The CONNECT record extension.....	203
The DELDSD record extension.....	205
The DELGROUP record extension.....	206
The DELUSER record extension.....	208
The PASSWORD record extension.....	209
The PERMIT record extension.....	211
The RALTER record extension.....	212
The RDEFINE record extension.....	214
The RDELETE record extension.....	215
The REMOVE record extension.....	217
The SETROPTS record extension.....	218
The RVARY record extension.....	220
The APPCLU record extension.....	221
The general event record extension.....	223
The directory search record extension.....	224
The check directory access record extension.....	227
The check file access record extension.....	229
The change audit record extension.....	231
The change directory record extension.....	234
The change file mode record extension.....	236
The change file ownership record extension.....	239
The clear SETID bits record extension.....	241
The EXEC SETUID/SETGID record extension.....	243
The GETPSENT record extension.....	245
The initialize z/OS UNIX record extension.....	247
The z/OS UNIX process completion record.....	249
The KILL record extension.....	250

The LINK record extension.....	252
The MKDIR record extension.....	254
The MKNOD record extension.....	257
The mount file system record extension.....	260
The OPENFILE record extension.....	262
The PTRACE record extension.....	265
The rename file record extension.....	267
The RMDIR record extension.....	269
The SETEGID record extension.....	271
The SETEUID record extension.....	273
The SETGID record extension.....	275
The SETUID record extension.....	276
The SYMLINK record extension.....	278
The UNLINK record extension.....	280
The unmount file system record extension.....	282
The check file owner record extension.....	284
The check privilege record extension.....	286
The open subsidiary TTY record extension.....	287
The RACLINK command record extension.....	289
The IPCCHK record extension.....	292
The IPCGET record extension.....	294
The IPCCTL record extension.....	296
The SETGROUP record extension.....	299
The CKOWN2 record extension.....	301
The access rights record extension.....	303
The RACDCERT command record extension.....	304
The InitACEE record extension.....	306
The Network Authentication Service record extension.....	308
The RPKIGENC record extension.....	308
The RPKIEXPT record extension.....	310
The Policy Director Authorization Services record extension.....	312
The RPKIREAD record extension.....	313
The RPKIUPDR record extension.....	315
The RPKIUPDC record extension.....	317
The SETFACL record extension.....	318
The DELFACL record extension.....	321
The SETFSECL record extension.....	323
The WRITEDWN record extension.....	324
The PKIDPUBR record extension.....	326
The RPKIRESP record extension.....	326
The PassTicket evaluation (PTEVAL) record extension.....	327
The PassTicket generation (PTCREATE) record extension.....	329
The RPKISCEP record extension.....	331
The RDATAUPD record extension.....	333
The PKIAURNW record extension.....	335
The PGMVERIFY record extension.....	336
The RACMAP record extension.....	337
The AUTOPROF record extension.....	339
The RPKIQREC record extension.....	341
The PKIGENC record extension.....	342
The PRLIMIT record extension.....	343
Security event record extension (unloaded).....	345

Chapter 7. The format of the unloaded SMF type 81 data..... 347

The format of the unloaded SMF type 81 class data.....	351
--	-----

Chapter 8. The format of the unloaded SMF type 83 data..... 353

Chapter 9. RACF database unload utility (IRRDBU00) records.....	355
IRRDBU00 record types.....	355
Format of the record type identification number.....	359
The relationships among unloaded database records.....	360
Conversion rules of the database unload utility.....	366
Record formats produced by the database unload utility.....	367
Group record formats.....	367
User record formats.....	370
Data set record formats.....	392
General resource record formats.....	397
Chapter 10. The RACF PassTicket.....	419
Generating and evaluating a PassTicket.....	419
Using the RCVTPTGN service to generate a PassTicket.....	420
Incorporating the PassTicket generator algorithm into your program.....	421
Generating a secured signon session key.....	433
Using the service to generate a secured signon session key.....	433
Incorporating the secured signon session key generator algorithm into your program.....	435
Chapter 11. The RACF environment service.....	437
Function.....	437
Requirements.....	437
RACF authorization.....	437
Register usage.....	437
Format.....	438
Parameters.....	438
Return and reason codes.....	440
Usage notes.....	441
Related services.....	442
Chapter 12. SAF user mapping plug-in interface.....	443
Installation considerations for the SAF user mapping service.....	443
Writing an application that uses the SAF user mapping plug-in interface.....	443
Preparing to run an application with the SAF user mapping plug-in implementation.....	445
Writing your own SAF user mapping plug-in implementation.....	446
SAF user mapping plug-in initialization function – safMappingInit().....	448
Function.....	448
Format.....	448
Requirements.....	448
RACF authorization.....	448
Usage notes.....	449
Function return values.....	450
SAF user mapping plug-in lookup function – safMappingLookup().....	450
Function.....	450
Format.....	450
Requirements.....	451
RACF authorization.....	451
Usage notes.....	451
Function return values.....	453
SAF user mapping plug-in termination function – safMappingTerm().....	454
Function.....	454
Format.....	455
Requirements.....	455
RACF authorization.....	455
Usage notes.....	455
Function return values.....	456

The irrspim.h header file.....	457
Chapter 13. Generic name translate service (IRRGNT00).....	461
Function.....	461
Environment.....	461
Restrictions.....	461
Input register information.....	461
Output register information.....	461
Parameters.....	462
Invoking the generic name translate service.....	462
Return and reason codes.....	463
Chapter 14. IRRXUTIL: REXX interface to R_admin extract.....	465
Parameters.....	465
Examples.....	469
Return codes.....	469
SETROPTS data.....	472
Specifying a period in the stem name.....	472
Examples.....	473
REXX stem variables created by IRRXUTIL for profile and SETROPTS information.....	473
Example.....	475
REXX stem variables created by IRRXUTIL for RACF subsystem and remote sharing information.....	476
Class descriptor entry data.....	485
Appendix A. ICHEINTY, ICHETEST, and ICHEACTN macros.....	489
ICHEINTY macro.....	490
Return codes from the ICHEINTY macro.....	502
ICHETEST macro.....	505
ICHEACTN macro.....	508
Using ICHEACTN with the DATAMAP=NEW and DATAMAP=OLD operands.....	511
Examples of ICHEINTY, ICHETEST, and ICHEACTN macro usage.....	516
Appendix B. REXX RACVAR.....	521
Appendix C. Supplied class descriptor table entries.....	523
Supplied class descriptor table entries.....	523
Appendix D. RACF database templates.....	607
Format of field definitions.....	607
Repeat groups on the RACF database.....	608
Field length.....	608
Data field types.....	608
Combination fields on the RACF database.....	609
Determining space requirements for the profiles.....	609
Determining space requirements for alias index entries.....	611
Group template for the RACF database.....	611
User template for the RACF database.....	614
Connect template for the RACF database.....	629
Data set template for the RACF database.....	631
General template for the RACF database.....	636
Reserved template for the RACF database.....	652
Appendix E. Event code qualifier descriptions.....	655
Event codes and event code qualifiers.....	655
Event 1(1): JOB INITIATION/TSO LOGON/TSO LOGOFF.....	655
Event 2(2): RESOURCE ACCESS.....	659

Event 3(3): ADDVOL/CHGVOL.....	661
Event 4(4): RENAME RESOURCE.....	662
Event 5(5): DELETE RESOURCE.....	663
Event 6(6): DELETE ONE VOLUME OF A MULTIVOLUME RESOURCE.....	663
Event 7(7): DEFINE RESOURCE.....	664
Event 8(8)–25(19): COMMANDS.....	665
Event 26(1A): APPCLU.....	666
Event 27(1B): GENERAL AUDITING.....	667
Event 28(1C)–58(3A): z/OS UNIX EVENT TYPES.....	667
Event 59(3B): RACLINK EVENT TYPES.....	671
Event 60(3C)–62(3E): z/OS UNIX XPG4 EVENT TYPES.....	672
Event 63(3F): z/OS UNIX SETGROUPS EVENT TYPE.....	672
Event 64(40): X/OPEN SINGLE UNIX SPECIFICATION EVENT TYPES.....	673
Event 65(41): z/OS UNIX PASSING OF ACCESS RIGHTS EVENT TYPES.....	673
Event 66(42)–67(43): CERTIFICATE EVENT TYPES.....	673
Event 68(44): GRANT OF INITIAL KERBEROS TICKET.....	674
Event 69(45): R_PKIServ GENCERT.....	674
Event 70(46): R_PKIServ EXPORT.....	674
Event 71(47): POLICY DIRECTOR ACCESS CONTROL DECISION.....	675
Event 72(48): R_PKIServ QUERY.....	675
Event 73(49): R_PKIServ UPDATEREQ.....	675
Event 74(4A): R_PKIServ UPDATECERT.....	675
Event 75(4B): CHANGE FILE ACL.....	676
Event 76(4C): REMOVE FILE ACL.....	676
Event 77(4D): SET FILE SECURITY LABEL.....	676
Event 78(4E): SET WRITE-DOWN PRIVILEGE.....	676
Event 79(4F): CRL PUBLICATION.....	676
Event 80(50): R_PKIServ RESPOND.....	677
Event 81(51): PassTicket Evaluation.....	677
Event 82(52): PassTicket Generation.....	677
Event 83(53): R_PKIServ SCEPREQ.....	677
Event 84(54): R_Datalib RDATAUPD.....	677
Event 85(55): PKIAURNW.....	678
Event 86(56): R_PgmSignVer.....	678
Event 87(57): RACMAP.....	678
Event 88(58): AUTOPROF.....	679
Event 89(59): RPKIQREC.....	679
Event 90(5A): PKIGENC.....	679
Appendix F. Accessibility.....	681
Notices.....	683
Terms and conditions for product documentation.....	684
IBM Online Privacy Statement.....	685
Policy for unsupported hardware.....	685
Minimum supported hardware.....	685
Programming interface information.....	686
Trademarks.....	686
Index.....	687

Figures

1. Relationship among the group record types.....	361
2. Relationship among the user record types (Part 1 of 2).....	362
3. Relationship among the user record types (Part 2 of 2).....	363
4. Relationship among the data set record types.....	364
5. Relationship among the general resource record types.....	365
6. RACF legacy PassTicket generator for secured signon.....	422
7. Algorithm for RACF legacy PassTicket time-coder used for secured signon.....	423
8. RACF enhanced PassTicket generator algorithm.....	424
9. RACF enhanced PassTicket time-coder algorithm.....	425
10. Permutation tables for RACF secured signon.....	429
11. Translation table for RACF secured signon.....	429
12. Translation table for RACF secured signon.....	432
13. Secured signon session key generation logic.....	435
14. CDMF key-weakening logic.....	436
15. SAF user mapping using EIM.....	444
16. The irrspim.h header file.....	457
17. The irrspim.h header file (cont.).....	458
18. The irrspim.h header file (cont.).....	459
19. The irrspim.h header file (cont.).....	459

Tables

1. RACROUTE request types and corresponding independent RACF system macro.....	31
2. Format of the SMF type 80 record.....	34
3. Table of extended-length relocate section variable data.....	65
4. Initialization record (type 81).....	136
5. RACF SMF type 83 record product section.....	147
6. RACF SMF record relocate section format.....	155
7. RACF SMF type 83 subtype 2 and above relocates.....	155
8. Format of the header portion of the unloaded SMF records.....	172
9. Event codes and descriptions.....	174
10. Format of the job initiation record extension (event code 01).....	177
11. Event qualifiers for JOBINIT records.....	180
12. Format of the ACCESS record extension (event code number 02).....	182
13. Event qualifiers for access records.....	185
14. Format of the ADDVOL record extension (event code 03).....	185
15. Event qualifiers for add volume/change volume records.....	187
16. Format of the RENAMEDS record extension (event code 04).....	187
17. Event qualifiers for RENAMEDS records.....	188
18. Format of the DELRES record extension (event code 05).....	189
19. Event qualifiers for delete resource records.....	190
20. Format of the DELVOL record extension (event code 06).....	191
21. Event qualifiers for delete volume records.....	192
22. Format of the DEFINE record extension (event code 07).....	192
23. Event qualifiers for define resource records.....	193

24. Format of the ADDSD record extension (event code 08).....	194
25. Event qualifiers for ADDSD command records.....	195
26. Format of the ADDGROUP record extension (event code 09).....	196
27. Event qualifiers for ADDGROUP command records.....	197
28. Format of the ADDUSER record extension (event code 10).....	197
29. Event qualifiers for ADDUSER command records.....	198
30. Format of the ALTDSD record extension (event code 11).....	199
31. Event qualifiers for ADDSD command records.....	200
32. Format of the ALTGROUP record extension (event code 12).....	200
33. Event qualifiers for ALTGROUP command records.....	202
34. Format of the ALTUSER record extension (event code 13).....	202
35. Event qualifiers for ALTUSER command records.....	203
36. Format of the CONNECT record extension (event code 14).....	203
37. Event qualifiers for CONNECT command records.....	205
38. Format of the DELDSD record extension (event code 15).....	205
39. Event qualifiers for DELDSD command records.....	206
40. Format of the DELGROUP record extension (event code 16).....	207
41. Event qualifiers for DELGROUP commands records.....	208
42. Format of the DELUSER record extension (event code 17).....	208
43. Event qualifiers for DELUSER command records.....	209
44. Format of the PASSWORD record extension (event code 18).....	209
45. Event qualifiers for PASSWORD command records.....	211
46. Format of the PERMIT record extension (event code 19).....	211
47. Event qualifiers for PERMIT command records.....	212
48. Format of the RALTER record extension (event code 20).....	212

49. Event qualifiers for RALTER command records.....	214
50. Format of the RDEFINE record extension (event code 21).....	214
51. Event qualifiers for RDEFINE command records.....	215
52. Format of the RDELETE record extension (event code 22).....	216
53. Event qualifiers for RDELETE command records.....	217
54. Format of the REMOVE record extension (event code 23).....	217
55. Event qualifiers for REMOVE command records.....	218
56. Format of the SETROPTS record extension (event code 24).....	219
57. Event qualifiers for SETROPTS command records.....	220
58. Format of the RVARY record extension (event code 25).....	220
59. Event qualifiers for RVARY command records.....	221
60. Format of the APPCLU record extension (event code 26).....	221
61. Event qualifiers for APPC session establishment records.....	223
62. Format of the general event record extension (event code 27).....	223
63. Format of the directory search record extension (event code 28).....	224
64. Event qualifiers for directory search records.....	227
65. Format of the check directory access record extension (event code 29).....	227
66. Event qualifiers for check directory records.....	229
67. Format of the check file access record extension (event code 30).....	229
68. Event qualifiers for check file access records.....	231
69. Format of the change audit record extension (event code 31).....	231
70. Event qualifiers for change audit records.....	234
71. Format of the change directory record extension (event code 32).....	234
72. Event qualifiers for change directory records.....	236
73. Format of the change file mode record extension (event code 33).....	236

74. Event qualifiers for change file mode records.....	239
75. Format of the change file ownership record extension (event code 34).....	239
76. Event qualifiers for change file ownership records.....	241
77. Format of the clear SETID bits record extension (event code 35).....	241
78. Event qualifiers for clear SETID records.....	243
79. Format of the EXEC with SETUID/SETGID record extension (event code 36).....	243
80. Event qualifiers for EXEC with SETID/SETGID records.....	245
81. Format of the GETPSENT record extension (event code 37).....	245
82. Event qualifiers for GETPSENT records.....	247
83. Format of the initialize z/OS UNIX process record extension (event code 38).....	247
84. Event qualifiers for initialize z/OS UNIX process records.....	248
85. Format of the z/OS UNIX process complete record extension (event code 39).....	249
86. Event qualifiers for z/OS UNIX process complete records.....	250
87. Format of the KILL process record extension (event code 40).....	250
88. Event qualifiers for KILL records.....	252
89. Format of the LINK record extension (event code 41).....	252
90. Event qualifiers for LINK records.....	254
91. Format of the MKDIR record extension (event code 42).....	254
92. Event qualifiers for MKDIR records.....	257
93. Format of the MKNOD record extension (event code 43).....	257
94. Event qualifiers for MKNOD records.....	260
95. Format of the mount file system record extension (event code 44).....	260
96. Event qualifiers for mount file system records.....	262
97. Format of the OPENFILE record extension (event code 45).....	262
98. Event qualifiers for OPENFILE records.....	265

99. Format of the PTRACE record extension (event code 46).....	265
100. Event qualifiers for PTRACE records.....	267
101. Format of the rename file record extension (event code 47).....	267
102. Event qualifiers for rename file records.....	269
103. Format of the RMDIR record extension (event code 48).....	269
104. Event qualifiers for RMDIR records.....	271
105. Format of the SETEGID record extension (event code 49).....	271
106. Event qualifiers for SETEGID records.....	273
107. Format of the SETEUID record extension (event code 50).....	273
108. Event qualifiers for SETEUID records.....	274
109. Format of the SETGID record extension (event code 51).....	275
110. Event qualifiers for SETGID records.....	276
111. Format of the SETUID record extension (event code 52).....	277
112. Event qualifiers for SETUID records.....	278
113. Format of the SYMLINK record extension (event code 53).....	278
114. Event qualifiers for SYMLINK records.....	280
115. Format of the UNLINK record extension (event code 54).....	280
116. Event qualifiers for UNLINK records.....	282
117. Format of the unmount file system record extension (event code 55).....	282
118. Event qualifiers for unmount file system records.....	284
119. Format of the check file owner record extension (event code 56).....	284
120. Event qualifiers for check file owner records.....	285
121. Format of the check privileges record extension (event code 57).....	286
122. Event qualifiers for check privileges records.....	287
123. Format of the open subsidiary TTY record extension (event code 58).....	287

124. Event qualifiers for open subsidiary TTY records.....	289
125. Format of the RACLINK command record extension (event code 59).....	289
126. Event qualifiers for RACLINK command records.....	292
127. Format of the IPCCHK record extension (event code 60).....	292
128. Event qualifiers for check IPC records.....	294
129. Format of the IPCGET record extension (event code 61).....	294
130. Event qualifiers for IPCGET records.....	296
131. Format of the IPCCTL record extension (event code 62).....	296
132. Event qualifiers for IPCCTL records.....	299
133. Format of the SETGROUP record extension (event code 63).....	299
134. Event qualifiers for SETGROUP records.....	301
135. Format of the CKOWN2 record extension (event code 64).....	301
136. Event qualifiers for CKOWN2 records.....	303
137. Format of the access rights record extension (event code 65).....	303
138. Event qualifiers for access rights records.....	304
139. Format of the RACDCERT command extension (event code 66).....	304
140. Event qualifiers for RACDCERT command records.....	306
141. Format of the InitACEE record extension (event code 67).....	306
142. Event qualifiers for InitACEE records.....	307
143. Format of the Network Authentication Service record extension (event code 68).....	308
144. Event qualifiers for Network Authentication Service records.....	308
145. Format of the RPKIGENC record extension (event code 69).....	308
146. Event qualifiers for RPKIGENC records.....	310
147. Format of the RPKIEXPT record extension (event code 70).....	311
148. Event qualifiers for RPKIEXPT records.....	312

149. Format of Policy Director Authorization Services record extension (event code 71).....	312
150. Event qualifiers for Policy Director Authorization Services records.....	312
151. Format of the RPKIREAD record extension (event code 72).....	313
152. Event qualifiers for RPKIREAD records.....	315
153. Format of the RPKIUPDR record extension (event code 73).....	315
154. Event qualifiers for RPKIUPDR records.....	317
155. Format of the RPKIUPDC record extension (event code 74).....	317
156. Event qualifiers for RPKIUPDC records.....	318
157. Format of the SETFACL record extension (event code 75).....	318
158. Event qualifiers for SETFACL records.....	320
159. Format of the DELFACL record extension (event code 76).....	321
160. Event qualifiers for DELFACL records.....	322
161. Format of the SETFSECL record extension (event code 77).....	323
162. Event qualifiers for SETFSECLrRecords.....	324
163. Format of the WRITEDWN record extension (event code 78).....	324
164. Event qualifiers for WRITEDWN records.....	325
165. Format of the PKIDPUBR record extension (event code 79).....	326
166. Event qualifiers for PKIDPUBR records.....	326
167. Format of the RPKIRESP record extension (event code 80).....	326
168. Event qualifiers for RPKIRESP records.....	327
169. Format of the PassTicket evaluation record extension (event code 81).....	328
170. Event qualifiers for PTEVAL records.....	329
171. Format of the PassTicket generation record extension (event code 82).....	329
172. Event qualifiers for PTCREATE records.....	331
173. Format of the RPKISCEP record extension (event code 83).....	331

174. Event qualifiers for RPKISCEP records.....	333
175. Format of the RDATAUPD record extension (event code 84).....	333
176. Event qualifiers for RDATAUPD records.....	334
177. Format of the PKIAURNW record extension (event code 85).....	335
178. Event qualifiers for PKIAURNW records.....	335
179. Format of the PGMVERYF record extension (event code 86).....	336
180. Event qualifiers for PGMVERYF records.....	337
181. Format of the RACMAP record extension (event code 87).....	337
182. Event qualifiers for RACMAP records.....	339
183. Format of the AUTOPROF record extension (event code 88).....	339
184. Event qualifiers for AUTOPROF records.....	340
185. Format of the RPKIQREC record extension (event code 89).....	341
186. Event qualifiers for RPKIQREC records.....	342
187. Format of the PKIGENC record extension (event code 90).....	342
188. Event qualifiers for PKIGENC records.....	343
189. Format of the prlimit record extension (event code 91).....	343
190. Event qualifiers for PRLIMIT records.....	344
191. Format of the security event record extension (event code 92).....	345
192. Event qualifiers for security event records.....	345
193. Format of the unloaded SMF type 81 records.....	347
194. Format of the unloaded SMF type 81 class records.....	351
195. Format of the unloaded SMF type 83 records.....	353
196. Group Basic Data Record.....	368
197. Group Subgroups Record.....	368
198. Group Members Record.....	368

199. Group Installation Data Record.....	369
200. Group DFP Data Record.....	369
201. Group OMVS Data Record.....	369
202. Group OVM Data Record.....	370
203. Group TME Data Record.....	370
204. Group CSDATA custom fields record.....	370
205. User basic data record.....	371
206. User Categories Record.....	374
207. User Classes Record.....	374
208. User Group Connections Record.....	374
209. User Installation Data Record.....	374
210. User Connect Data Record.....	375
211. User RRSF Data Record.....	375
212. User Certificate Name Record.....	376
213. User Associated Mappings Record.....	376
214. User Associated Distributed Mappings Record.....	377
215. User MFA factor data record.....	377
216. User MFA policies record.....	377
217. User DFP Data Record.....	378
218. User TSO data record.....	378
219. User CICS data record.....	379
220. User CICS operator class record.....	379
221. User CICS RSL key record.....	379
222. User CICS TSL Key Record.....	379
223. User language data record.....	380

224. User OPERPARM data record.....	380
225. User OPERPARM scope record.....	387
226. User WORKATTR data record.....	387
227. User OMVS data record.....	388
228. User NETVIEW segment record.....	388
229. User OPCLASS record.....	389
230. User DOMAINS record.....	389
231. User DCE data record.....	389
232. User OVM data record.....	389
233. User LNOTES data record.....	390
234. User NDS data record.....	390
235. User KERB data record.....	390
236. User PROXY record.....	391
237. User EIM Record.....	391
238. User CSDATA Custom fields record.....	392
239. User MFA factor tags data record.....	392
240. Data Set Basic Data Record.....	392
241. Data Set Categories Record.....	394
242. Data Set Conditional Access Record.....	394
243. Data Set Volumes Record.....	395
244. Data Set Access Record.....	395
245. Data Set Installation Data Record.....	395
246. Data Set Member Record.....	396
247. Data Set DFP Data Record.....	396
248. Data Set TME Data Record.....	397

249. Data set CSDATA record.....	397
250. General Resource Basic Data Record.....	399
251. General Resource Tape Volume Record.....	400
252. General Resource Categories Record.....	401
253. General Resource Members Record.....	401
254. General Resource Volumes Record.....	402
255. General Resource Access Record.....	403
256. General Resource Installation Data Record.....	403
257. General Resource Conditional Access Record.....	403
258. General Resource Filter Data Record.....	404
259. General Resource Distributed Identity Mapping Record.....	404
260. General Resource Session Data Record.....	405
261. General Resource Session Entity Record.....	405
262. General Resource DLF Data Record.....	406
263. General Resource DLF Job Names Record.....	406
264. General Resource SSIGNON Data Record.....	406
265. General Resource Started Task Data Record.....	407
266. General Resource SystemView Data Record.....	407
267. General Resource Certificate Data Record.....	407
268. General Resource Certificate References Record.....	408
269. General Resource Key Ring Data Record.....	408
270. General Resource TME Data Record.....	409
271. General Resource TME Child Record.....	409
272. General Resource TME Resource Record.....	409
273. General Resource TME Group Record.....	410

274. General Resource TME Role Record.....	410
275. General Resource KERB Data Record.....	410
276. General Resource PROXY Record.....	411
277. General Resource EIM Record.....	412
278. General Resource Alias Data Record.....	412
279. General Resource CDTINFO Data Record.....	412
280. General Resource ICTX Data Record.....	413
281. General Resource CFDEF Data Record.....	414
282. General Resource SIGVER Data Record.....	414
283. General Resource ICSF Record.....	415
284. General Resource ICSF key label Record.....	415
285. General Resource ICSF certificate identifier Record.....	416
286. General resource MFA factor definition record.....	416
287. General resource MFAPOLICY definition record (05I0).....	416
288. General resource MFA policy factors record (05I1).....	417
289. General resource CSDATA record (05J1).....	417
290. General resource IDTPARMS definition record (05k0).....	417
291. General Resource JES Data Record.....	418
292. General resource certificate information record.....	418
293. SAF user mapping plug-in dlls, header files, and side deck	443
294. The parameters of the safMappingInit() function.....	449
295. The SAF return values and the plug-in reason codes for the safMappingInit() function.....	450
296. The parameters of the safMappingLookup() function.....	451
297. The SAF return values and the plug-in reason codes for the safMappingLookup() function	454
298. The parameters of the safMappingTerm() function.....	455

299. The SAF return values and the plug-in reason codes for the safMappingTerm() function	456
300. IRRXUTIL parameters.....	466
301. Variables to alter the behavior of IRRXUTIL.....	469
302. Return codes.....	469
303. REXX stem variables.....	473
304. RRSF extract stem variables.....	476
305. Class descriptor entry data.....	485
306. ICHEINTY parameters.....	495
307. Format for the ICHEINTY work area when INDEX=MULTIPLE is not specified.....	499
308. Format for the ICHEINTY work area when INDEX=MULTIPLE is specified.....	499
309. ICHETEST parameters.....	507
310. ICHEACTN parameters.....	511
311. Classes supplied by IBM.....	523

About this document

This document supports z/OS (5655-ZOS) and describes Resource Access Control Facility (RACF®), which is part of z/OS Security Server.

Purpose of this document

This document contains a description (including syntax and related information) of macros provided with RACF. In addition, this publication provides information on coding the interfaces used to invoke RACF from the RACF ISPF panels.

It does not document the RACROUTE macro and the independent RACF system macros (such as RACHECK, RACDEF, and RACINIT) that are documented in *z/OS Security Server RACROUTE Macro Reference*. RACF callable services and their associated data areas are documented in *z/OS Security Server Callable Services*.

Who should use this document

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

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

Readers should be familiar with the information in *z/OS Security Server RACF Security Administrator's Guide*, *z/OS Security Server RACROUTE Macro Reference*, and *z/OS Security Server RACF System Programmer's Guide*.

z/OS Security Server RACF Auditor's Guide, which describes the RACF report writer, might also be useful.

How to use this document

The major sections of this document contain information on the RACF product macros and interface information. Each description includes:

- A general description of the service that the macro performs,
- A table of syntax rules that you must follow when you code the macro,
- A list of the parameters you can specify and an explanation of each parameter.
- **Chapter 1, “RACF customization macros,” on page 1**, provides information on macros that are provided by RACF. System programmers can use these macros to tailor RACF to meet the needs of your installation in various ways.
- **Chapter 2, “Panel driver interface module (ICHSPF03),” on page 21**, describes how installations can implement a panel driver interface between an application program and the RACF panels.
- **Chapter 3, “Profile name list service routine (IRRPNL00),” on page 25**, describes how installations using TSO/E can use IRRPNL00 to call RACF to retrieve the names of profiles.
- **Chapter 4, “Date conversion routine,” on page 29**, describes the Date Conversion Routine, how to invoke it and the format of a Returned Converted Date.
- **Chapter 5, “SMF records,” on page 31**, contains information on SMF record types 80, 81, and 83 including reformatted RACF SMF records.
- **Chapter 6, “The format of the unloaded SMF type 80 data,” on page 169, Chapter 7, “The format of the unloaded SMF type 81 data,” on page 347, and Chapter 8, “The format of the unloaded SMF**

type 83 data,” on page 353 contain detailed descriptions of the records that are produced by the RACF SMF data unload utility.

- **Chapter 9, “RACF database unload utility (IRRDBU00) records,” on page 355**, contains detailed descriptions of the records that are produced by the RACF database unload utility.
- **Chapter 10, “The RACF PassTicket,” on page 419**, describes the PassTicket, an alternative to the RACF password.
- **Chapter 11, “The RACF environment service,” on page 437**, contains the Environment Service functions, descriptions, requirements, and parameters.
- **Chapter 12, “SAF user mapping plug-in interface,” on page 443**, describes the C functions that comprise the SAF user ID mapping service and provides implementation information for applications that use them.
- **Appendix A, “ICHEINTY, ICHETEST, and ICHEACTN macros,” on page 489**, provides information on the ICHEINTY macro and the macros that work as part of it: ICHETEST and ICHEACTN. Very experienced programmers who want to write their own code to interface with the RACF database can use these macros to do so. However, due to the complexity of the ICHEINTY macro, most programmers use the RACROUTE REQUEST=EXTRACT macro instead.
- **Appendix B, “REXX RACVAR,” on page 521**, describes the RACF service for REXX EXECs that provides information about the running user.
- **“Supplied class descriptor table entries” on page 523**, lists the class entries that are supplied by IBM in the class descriptor table (ICHRRCDX).
- **Appendix D, “RACF database templates,” on page 607**, contains database templates.
- **Appendix E, “Event code qualifier descriptions,” on page 655**, contains detailed explanations of the SMF event code qualifiers.

z/OS information

This information explains how z/OS references information in other documents and on the web.

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

Summary of changes

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

Note: IBM z/OS policy for the integration of service information into the z/OS product documentation library is documented on the z/OS Internet Library under [IBM z/OS Product Documentation Update Policy](http://www.ibm.com/docs/en/zos/latest?topic=zos-product-documentation-update-policy) (www.ibm.com/docs/en/zos/latest?topic=zos-product-documentation-update-policy).

Summary of changes for z/OS 3.2

The following content is new, changed, or no longer included in z/OS 3.2.

New

The following content is new.

September 2025 release

- **RACF user ID containment:** RACF provides the ability to contain a user ID, which immediately stops the user from accessing RACF-protected resources, even during an active session. The user ID containment functionality, which is an extension of RACF user ID revocation processing, is available in z/OS 3.2 and z/OS 3.1 when you install the PTF for [APAR OA67286](http://www.ibm.com/support/pages/apar/OA67286) (www.ibm.com/support/pages/apar/OA67286).

For the updates related to this function, see the following sections:

- New event type 92 is defined for the SMF Type 80 record. See [“Table of event codes and event code qualifiers”](#) on page 41.
- New relocate section 68 is defined for the SMF Type 80 record. See [“Security event record extension \(unloaded\)”](#) on page 345.
- New fields to describe the user containment status are defined for the Database Unload (IRRDBU00) user basic record. See [“User basic data record \(0200\)”](#) on page 371.
- **RACF RACDCERT certificate generation support of multiple alternate names:** For the SMF type 80 record, in the RACF processing record subsection (SMF80DA2), the descriptions of several fields are revised. See [“Table of extended-length relocate section variable data”](#) on page 65.
- **DSNRAUTH class:** The DSNRAUTH class is added to the SETROPTS RACLIST keyword. For more information, see [Table 311](#) on page 523.

Changed

The following content is changed.

September 2025 release

- None.

Deleted

The following content is deleted.

September 2025 release

- None.

Summary of changes for z/OS 3.1

The following content is new, changed, or no longer included in z/OS 3.1.

New

The following content is new.

June 2025

- Relocate type section 67 has been added to events 81 and 82 in [“Table of event codes and event code qualifiers”](#) on page 41.

April 2025

- In support of APARs OA65299 and OA66783, information about using identity tokens (IDTs) with RSA signatures is added to the following topics:
 - [“Table of extended-length relocate section variable data”](#) on page 65
 - Fields are added to the SMF type 80 event code 1 (RACROUTE REQ=VERIFY/X) record. See [“Table of extended-length relocate section variable data”](#) on page 65.
 - [“The JOBINIT record extension”](#) on page 177
 - [“General resource IDTPARMS definition record \(05k0\)”](#) on page 417

March 2025

- The ENCTYPES field has been added to the DFP segment of the data set template described in [“Data set template for the RACF database”](#) on page 631 and [“Data set DFP data record \(0410\)”](#) on page 396.

February 2024

- In SMF Record Type 80, Relocate Section 6 (command-related data), a bit is defined for the new KDFAES keyword of the SETROPTS command. This support is added by APAR OA65423, which also applies to z/OS 2.5 and z/OS 2.4. See the SETROPTS section in [“Table of data type 6 command-related data”](#) on page 81.
- In SMF Record Type 80, in the SMF80REA field ("Reason for logging"), explanations are improved for bit 3 and bit 7 (APAR OA63211). See Table Note [“2”](#) on page 40.

January 2024

A new flag bit PBEAES is set when the PBE(AES) keyword is specified on the RACDCERT EXPORT command (APAR OA65002). See [“Table of data type 6 command-related data”](#) on page 81.

October 2023

- PRLIMIT is added to the table of event codes and event code qualifiers; see [“Table of event codes and event code qualifiers”](#) on page 41.
- PRLIMIT is a new record extension; see [“The PRLIMIT record extension”](#) on page 343.

September 2023 release

- A new field, GRCFDEF_ACEE, is added to the General Resource CFDEF Data record, which defines custom field information. See [“General Resource CFDEF Data record \(05E0\)”](#) on page 414.
- A new field, CFACEE, is added to the CFDEF segment of the GENERAL template. See [“General template for the RACF database”](#) on page 636.
- SETROPTS APPLAUDIT is extended to log successful logons. The OPTAUDIT class is added to the list of supplied classes. For more information, see [Appendix C, “Supplied class descriptor table entries,”](#) on page 523.

Changed

The following content is changed.

September 2023 release

- The description of SMF80VRM in the table of extended-length relocate section variable data is extended with a new value “77E0 z/OS Security Server (RACF) V3 R1.” See [“Format of SMF type 80 records” on page 33.](#)
- The description of SMF81VRM in the table of extended-length relocate section variable data is extended with a new value “77E0 z/OS Security Server (RACF) V3 R1.” See [“Record type 81: RACF initialization record” on page 136.](#)
- The description of SMF83VRM in the table of extended-length relocate section variable data is extended with a new value “77E0 z/OS Security Server (RACF) V3 R1.” See [“Subtype 1” on page 147.](#)

Deleted

The following content was deleted.

September 2023 release

- None.

Chapter 1. RACF customization macros

This topic contains information that is *not* a programming interface. It is intended to help the installations that use RACF product macros to customize a RACF installation.

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

- **“ICHERCDE macro” on page 1** used to generate entries for the static class descriptor table.
- **“ICHNCONV macro” on page 10** used to create the installation's naming convention table.
- **“ICHRFRTB macro” on page 19** used to generate entries in the RACF router table.

For the descriptions and functions of the ICHEINTY, ICHECTEST, and ICHEACTN product macros that can be used to locate, update, test, and retrieve various profiles in the RACF database see [Appendix A, “ICHEINTY, ICHECTEST, and ICHEACTN macros,” on page 489](#).

Guideline: Because of the complexity of the ICHEINTY, ICHECTEST, and ICHEACTN macros and the cautions required in their use, you should use the RACROUTE REQUEST=EXTRACT system macro instead. See [z/OS Security Server RACROUTE Macro Reference](#) for more information.

ICHERCDE macro

Guideline: If your installation needs to define resource classes, to avoid the need to re-IPL, do not define your classes in the static class descriptor table using the ICHERCDE macro. Instead, define your classes in the dynamic class descriptor table using RDEFINE and RALTER commands for the CDT resource class. For more information about the dynamic class descriptor table, see [z/OS Security Server RACF Security Administrator's Guide](#).

The *class descriptor table* contains information that directs the processing of general resources. The table consists of an entry for each resource class except USER, GROUP, and DATASET. The class descriptor table contains entries that IBM supplies, and, optionally, entries defined by the installation. It has two parts:

- The *static class descriptor table* contains the entries that IBM supplies (shipped in the module ICHRRCDX), and, optionally, entries defined by the installation (in the module ICHRRCDE). You must not change ICHRRCDX. To create or modify ICHRRCDE, use the ICHERCDE macro. You must re-IPL for the updates to ICHRRCDE to take effect.
- The *dynamic class descriptor table* contains entries built from the CDT general resource class. RACF treats the dynamic class descriptor table as a logical extension to the static class descriptor table. To create or modify the dynamic class descriptor table, use the RDEFINE and RALTER commands. You do not need to re-IPL for updates made to the dynamic class descriptor table to take effect.

Restriction: A grouping class in the dynamic class descriptor table cannot reference a member class in the static class descriptor table, and a member class in the dynamic class descriptor cannot reference a grouping class in the static class descriptor.

Note: The remainder of this topic contains information about the static class descriptor table. For information about the dynamic class descriptor table, see [z/OS Security Server RACF Security Administrator's Guide](#).

The ICHERCDE macro generates entries for the static class descriptor table. To generate the table, you must invoke the macro once for each class. To identify the end of the static class descriptor table, invoke the macro without specifying any operands.

The installation-defined static class descriptor table, module ICHRRCDE, must have RMODE(24) and must reside in SYS1.LINKLIB or another library in your linklist concatenation. Refer to [z/OS Security Server RACF System Programmer's Guide](#) for instructions on how to create ICHRRCDE.

Member RACINSTL in SYS1.SAMPLIB, contains among other items, a sample job stream for updating or creating a static installation-defined class descriptor table (ICHRRCDE).

Notes:

1. A maximum of 1024 classes can be defined in the class descriptor table. There are 1024 POSIT values, of which numbers 19–56 and 128–527 are available for installation use. Numbers 0–18, 57–127, and 528–1023 are reserved for IBM's use.
2. Installations sharing a database do not need identical class descriptor tables, but they must be compatible. If the same class is present on multiple systems, it must have the same attributes; for example, the POSIT numbers must be the same. Therefore, if systems X and Y are sharing a database, and system X has a class descriptor table with classes a, b, and c, and system Y has a class descriptor table with classes a, b, c, d, e, and f, the classes a, b, and c must be defined identically on both systems. However, system Y might have classes d, e, and f that are not defined on system X. Note that when RACF is enabled for sysplex communication, to allow flexibility when adding new classes to the class descriptor table RACF does not enforce consistency in the class descriptor table as it does with the data set name table and the range table.

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

The ICHERCDE macro definition is as follows:

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

CLASS=class name

Specifies the name of the resource class. The name must be 1–8 characters long and must consist of the following: A through Z, 0 through 9, or # (X'7B'), @ (X'7C'), \$ (X'5B'). The first character must be A through Z, # (X'7B'), @ (X'7C'), or \$ (X'5B'). You must include a # (X'7B'), @ (X'7C'), \$ (X'5B'), or numeric character in the name of any class you define to guarantee that installation-defined classes do not conflict with classes supplied by IBM. In this way, classes supplied by IBM should always have unique class names. If this rule is not followed, the assembler issues a severity 4 MNOTE warning.

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

Note: A class defined in the dynamic class descriptor can have 0 through 9 as the first character of its name. For more information about the dynamic class descriptor table, see *z/OS Security Server RACF Security Administrator's Guide*.

CASE=UPPER | ASIS

Specifies whether mixed-case profile names are allowed for the class specified by the CLASS operand. UPPER is the default. 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 specified in the FIRST= and OTHER= operands.

DFTRETC=0|4|8

Specifies the return code that RACF provides from RACROUTE REQUEST=AUTH, or REQUEST=FASTAUTH when RACF and the class are active and (if required) the class has been processed using SETROPTS RACLIST, but RACF does not find a profile to protect the resource specified on the AUTH or FASTAUTH request.

0

The access request was accepted.

4

No profile exists.

8

The access request was denied.

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

DFTUACC= 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. If you omit DFTUACC, and no access level is specified at the time the profile is created, RACF uses the default universal access authority from the command issuer's ACEE.

EQUALMAC=YES|NO

Specifies whether equal mandatory access checking is required when users attempt to access resources protected by profiles in this class. If EQUALMAC=YES is specified, whenever RACF performs a mandatory access check the security label of the user and the security label of the resource must be equivalent to pass the mandatory access check. Security labels are equivalent when they have the same security level and category definitions. The SYSMULTI security label is equivalent to any other security label.

Use EQUALMAC=YES for classes where two-way communication is expected.

EQUALMAC=YES cannot be specified with RVRSMAC=YES.

FIRST=

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

ALPHA

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

NUMERIC

Specifies a digit (0–9).

ALPHANUM

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

ANY

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

Notes:

1. Resource names (as opposed to profile names) for a class should not contain the characters *, %, or & because these characters do not work as expected when generic profile processing is active for the class.
2. This option includes the period (.), therefore, it is needed if you intend to use it as a delimiter.

NONATABC

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

NONATNUM

Specifies an alphabetic or numeric character. Characters such as # (X'7B'), @ (X'7C'), and \$ (X'5B') are excluded.

GENERIC=ALLOWED|DISALLOWED

Specifies whether SETROPTS GENERIC and SETROPTS GENCMD are to be allowed for the class. The SETROPTS GENERIC command activates generic profile checking for a class, and the SETROPTS GENCMD command activates generic profile command processing for a class.

If GENERIC=DISALLOWED is specified, GENLIST=ALLOWED cannot be specified.

Because generic processing is not allowed for grouping classes, GENERIC=ALLOWED cannot be specified if MEMBER= is also specified.

GENERIC keyword consideration for a class that shares a POSIT number:

- If the class shares a POSIT number with another class, all classes with the shared POSIT number must have the same setting for the GENERIC keyword. This is because the SETROPTS GENERIC and SETROPTS GENCMD commands process all classes that share a POSIT number.
- If your class that shares a POSIT number violates this rule, that is, at least one class specifies GENERIC=DISALLOWED and at least one class specifies GENERIC=ALLOWED, the assembler issues a severity 8 MNOTE error.
- If the class shares a POSIT number with an IBM class and it violates this rule, a warning message is issued during RACF initialization, and the value of the GENERIC keyword is changed by RACF to match the IBM class.
- If the class shares a POSIT number with another installation-defined class from a separate assembly and violates this rule, a warning message is issued during RACF initialization, and the value of the GENERIC keyword in both classes are set to the least restrictive attribute, GENERIC=ALLOWED.
- If your static installation class specifies GENERIC=DISALLOWED, and then a dynamic class is added that shares a POSIT number and specifies GENERIC=ALLOWED, the static class is changed to GENERIC=ALLOWED (the least restrictive attribute) during SETROPTS RACLIST(CDT) processing. The GENERIC=ALLOWED setting remains during that IPL.
- If you want to change the setting in the static class back to GENERIC=DISALLOWED, do the following tasks:
 1. Change the dynamic class to specify either a different POSIT number or GENERIC=DISALLOWED. See *z/OS Security Server RACF Security Administrator's Guide* for more guidelines for changing dynamic CDT entries.
 2. Re-IPL the system.

Exception: A grouping class and member class can share a POSIT number. GENERIC=DISALLOWED should be specified for the grouping class and GENERIC=ALLOWED may be specified 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, then if a user requests access to a resource protected by a generic profile, a copy of that profile is brought into the common storage area, rather than into the user's address space. RACF uses those generic profiles in common storage to check the authorization of any users who want to access the resource. The profiles remain in common storage until a REFRESH occurs.

GROUP=group-class

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

ID=number

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

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

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

KEYQUAL=nnn

Specifies the number of matching qualifiers RACF uses when loading generic profile names to satisfy an authorization request if a discrete profile does not exist for the resource. For example, if you specify two for the class, all generic profile names whose two highest level qualifiers match the two highest qualifiers of the entity name are loaded into the user's storage when the user requests access to a resource.

If you do not specify KEYQUAL, the default is 0, and profile names for the entire class are loaded and searched. The maximum value that you can specify for KEYQUAL is 123, which is the maximum number of qualifiers in a name 246 characters long.

When KEYQUAL=nnn is coded in the ICHERCDE macro, generic profiles that are created in that class might not contain generic characters in the first nnn qualifiers of the profile.

If a KEYQUAL=nnn is greater than 0 for a class, all discrete and generic profiles in that class must have at least nnn+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 nnn+1 qualifier or beyond.

Examples of valid profile names for KEYQUALIFIERS(2) are:

- A.B.C
- A.B.**
- A.B.C.D*

KEYQUAL=nnn (where nnn is greater than 0) should be used for a class that has the following attributes:

- The class is usually not RACLISTed.
- The class is usually not GENLISTed.
- Profile names in the class have a naming convention where many generic profiles have the same nnn qualifiers at the beginning of the profile name.

For example, suppose that you have an application program that uses an installation class to protect reports on terminal usage, and you have the following profiles for *every* 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.*
```

You might define your installation class with KEYQUAL=3 so that when an authorization check is done for a resource in your class, only the generic profile whose name matches the first three qualifiers of your report is loaded into storage for RACF to match against.

MAXLENX=number

Specifies the maximum length of resource and profile names for this class when a RACROUTE macro is invoked with the ENTITYX keyword, or a profile is added or changed using a RACF command processor. For installation-defined classes, you can specify a number from 1 to 246. If MAXLENX is not specified, the value specified for MAXLNTH is used.

Notes:

1. Do not assemble a static class descriptor table using MAXLENX and share it with a system running a RACF release earlier than OS/390® V2R8.
2. If you specify a MAXLENX value greater than the MAXLNTH value for a class before you define any profiles with names longer than MAXLNTH, verify that any programs using RACROUTE REQUEST=EXTRACT, TYPE=EXTRACTN or ICHEINTY NEXT for that class properly handle the longer names.

MAXLNTH=8|number

Specifies the maximum length of resource and profile names for this class when MAXLENX is not specified. When MAXLENX is also specified, MAXLNTH represents the maximum length of a resource name only when a RACROUTE macro is invoked with the ENTITY keyword. For installation-defined classes, you can specify a number from 1 to 246; the default is 8.

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

This restriction does *not* apply to transaction grouping classes.

MEMBER=member-class

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

OPER=YES|NO

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

OTHER=

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

ALPHA

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

NUMERIC

Specifies a digit (0–9).

ALPHANUM

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

ANY

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

Notes:

1. Resource names (as opposed to profile names) for a class should not contain the characters *, %, or & because these characters do not work as expected when generic profile processing is active for the class.
2. This option includes the period ('.'), therefore, it is needed if you intend to use it as a delimiter.

NONATABC

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

NONATNUM

Specifies an alphabetic or numeric character. Characters such as # (X'7B'), @ (X'7C'), and \$ (X'5B') are excluded.

POSIT=number

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

- 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 user has CLAUTH to a resource class
- Whether special resource access auditing applies to the class (SETROPTS LOGOPTIONS)
- Whether SETROPTS RACLIST occurs for this class (when the parameter RACLIST=ALLOWED is also coded)

Before you assemble the static class descriptor table (CDT), you must decide whether to use a unique set of option flags for each RACF class or whether to have two or more RACF classes share the same set of option flags.

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

Guidelines:

- A RACF class that has a default return code of 8 should not share a POSIT value with a RACF class having a default return code not equal to 8. If a class with a default return code of 8 is activated but no profiles are defined, user activity that requires access in that class is prevented.

There are 1024 POSIT numbers that can identify 1024 sets of option flags. Installations can specify POSIT numbers 19–56 and 128–527. Numbers 0–18, 57–127, and 528–1023 are reserved for IBM's use.

- If a class shares a POSIT number with another class, all classes with the shared POSIT number must have the same setting for the GENERIC keyword. See the GENERIC keyword for more information.

Note: The following text describes the use of POSIT numbers for classes in the static class descriptor table. You can add, delete, and change classes and change their POSIT numbers without the need for re-IPLing if you define your classes in the dynamic class descriptor table. For information about the dynamic class descriptor table, see *z/OS Security Server RACF Security Administrator's Guide*.

Adding a new class where a unique POSIT number is wanted to the static class descriptor table:

Suppose that you decide to define a new class called \$TSTCLAS. Since you want this class to be administered separately from any other class, you select a new POSIT number, 22, which is not being used by any other class. Now, when you activate or deactivate SETROPTS options for \$TSTCLAS, or grant CLAUTH to this class, no other classes are affected.

Adding a new class that shares a POSIT number with an existing class to the static class descriptor table: Suppose that you have a class that is called \$PONIES that was previously

defined with a unique POSIT number, 21. SETROPTS CLASSACT, SETROPTS AUDIT, and SETROPTS STATISTICS are currently in effect on your system for class \$PONIES as a result of issuing those commands for class \$PONIES.

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

When IPLing with the new ICHRRCDE, the same RACF processing options that are in effect for class \$PONIES are automatically in effect for the new class \$HORSES: SETROPTS CLASSACT, SETROPTS AUDIT, and SETROPTS STATISTICS.

Further, issuing *either* of the following commands:

- SETROPTS GLOBAL (\$PONIES)
- SETROPTS GLOBAL (\$HORSES)

activates global access checking for *both* the \$PONIES and the \$HORSES classes. Similarly, issuing *either* of the following commands:

- SETROPTS STATISTICS (\$PONIES)
- SETROPTS STATISTICS (\$HORSES)

activates STATISTICS for *both* the \$PONIES and the \$HORSES classes.

Any number of classes can share the same POSIT number. For example, a third class called \$MARES could be added and could also share POSIT number 21 with \$PONIES and \$HORSES. Sharing a POSIT number simplifies administration of related classes.

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

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

Changing an existing installation-defined class in the static class descriptor table: If you change the POSIT value, be aware that changing the POSIT value could cause unexpected results. For example, you could deactivate a class if you change it to use a POSIT value associated with a class that is not active.

If you are changing the POSIT value, do the following before making the change:

1. Issue the SETROPTS LIST command and record each active option for the class.
2. Examine your classes to see whether any other class is using the current POSIT value. If not, use the SETROPTS command to turn off all the options that are associated with the class so that you do not receive any extraneous options set if you later add a class using that POSIT value.
3. Change the POSIT number associated with the class by updating the ICHERCDE command for the class with the new POSIT number, re-creating ICHRRCDE, and re-IPLing all systems that use the class.
4. Use the SETROPTS command to set any of the options that are still relevant for the class, using the output of the previous SETROPTS LIST command as reference.

Deleting an installation-defined class from the static class descriptor table: You can delete a class entry from the static class descriptor table by specifying the name of the class to be deleted on the

OS-linkage-editor REPLACE statement. For the deletion to take effect, re-IPL all systems that used the class.

You should ensure that all profiles relating to this class are deleted *before* deleting the class descriptor table entry.

Pay special attention to any *unique* POSIT values you use. If the class you are deleting has a *unique* POSIT value, issue a SETROPTS LIST to check what options you are using with the class, for example, CLASSACT, LOGOPTIONS, AUDIT, RACLIST, and so on. Turn off each of the options for the class.

An example: You might have activated your class. You should deactivate the class before re-IPLing your system. If you do not deactivate the class and, at a future date, you create a class with the POSIT value previously used, the class will automatically be active. The same consideration applies to each option controlled by the POSIT value.

PROFDEF=YES|NO

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

RACLIST=ALLOWED|DISALLOWED

Specifies whether SETROPTS RACLIST is to be allowed for the class. If you process the 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.

RACLREQ=YES|NO

Specifies whether you must process the class using SETROPTS RACLIST to use RACROUTE REQUEST=AUTH. The purpose of this keyword is to allow routines that cannot tolerate I/O to invoke RACF. If you specify YES, and the class is not processed by SETROPTS RACLIST and a RACROUTE REQUEST=AUTH is attempted, the return code is 4. If you do not specify the parameter, it defaults to NO.

RVRSMAC=YES|NO

Specifies whether reverse mandatory access checking is required.

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

RVRSMAC=YES cannot be specified with EQUALMAC=YES.

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

SIGNAL=YES|NO

Specifies whether an ENF signal is sent to listeners when a SETROPTS RACLIST, SETROPTS NORACLIST, or SETROPTS RACLIST REFRESH is issued for the class, activating, deactivating, or updating the profiles used for authorization checking. For information about signals, see [ENF signals](#) in *z/OS Security Server RACF System Programmer's Guide*.

SLBLREQ=YES|NO

Specifies whether a security label is required for the profiles of this class.

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

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

ICHNCONV macro

RACF requires a data set name format where the high-level qualifier of a data set name is a RACF-defined user ID or group name. If your installation's data set naming convention already meets this requirement, you should not have to use this macro.

RACF allows installations to create a naming convention table (ICHNVC00) that RACF uses to check the data set name in all the commands and SVC routines that process data set names. This table helps an installation set up and enforce data set naming conventions that are different from the standard RACF naming conventions.

RACF compares a data set name against each entry in the table until it finds one that matches the name. If RACF does not find a matching entry, the name remains unchanged.

You create a naming convention table, ICHNVC00, by using the ICHNCONV macro. You must assemble the table and link-edit it into SYS1.LPALIB. ICHNVC00 is link-edited with AMODE(31) and RMODE(ANY).

The table can have up to 400 naming convention entries and can handle data set names of different formats. Each table entry consists of:

- One ICHNCONV DEFINE macro to start the naming convention and assign it a name
- Zero or more ICHNCONV SELECT macro to specify the conditions when the naming convention processes the data set name
- Zero or more ICHNCONV ACTION macro to convert the name to the standard RACF format or to change any of the modifiable variables
- One ICHNCONV END macro to terminate the naming convention

At the end of all the naming conventions, an ICHNCONV FINAL macro terminates the table itself.

ICHNCONV coding guideline

When writing a naming conventions table, be sure that your output data set names do not match any input data set names. If they match, you might receive unpredictable results. For example, suppose that you have erroneous entries in your naming conventions table that transform the following input data set names to the following output data set names:

Poor example:

```
A.B.#ANY.THING to B#.ANY.THING
B#.ANY.THING to C#.ANY.THING
```

When a data set named A.B.#ANY.THING is processed, it is transformed to B#.ANY.THING. If B#.ANY.THING is processed again as an input data set name, it is transformed again, this time to C#.ANY.THING. You should avoid this type of coding in your naming convention table.

This is important because you cannot predict the input data set names that are passed to the naming convention table. In some cases, the table might receive the input data set name in "external" format, as provided by a user, such as from a LISTDSD DA(*hlq.name*) command. In other cases, it might receive the input data set name in "internal" format, as previously processed by the naming convention table, such as from an authorization check made during processing of SEARCH or LISTDSD PREFIX(*hlq*) when RACF has retrieved the data set name from the database.

ICHNCONV DEFINE

An ICHNCONV DEFINE macro starts a naming convention and assigns it a name.

The format of the ICHNCONV DEFINE macro is:

```
[label] ICHNCONV DEFINE,NAME=convention name
```


DEFINE

Identifies the start of a naming convention. The ICHNCONV DEFINE must start each naming convention and there must be only one per convention.

NAME=convention name

Specifies a unique name that you can use for the convention.

The convention name must be 1 to 8 characters long and follow the rules for symbols in assembly language.

ICHNCONV SELECT

An ICHNCONV SELECT macro specifies the conditions when the naming convention processes the data set name.

The format of the ICHNCONV SELECT macro is:

```
[label] ICHNCONV  SELECT,COND=(condition|compound condition)
```

SELECT

Identifies that this convention has selection criteria.

If the condition on the COND parameter is true, the actions on the ICHNCONV ACTION macros are processed, and processing continues as specified on the ICHNCONV END macro. If the condition on the COND parameter is not true, RACF bypasses the ICHNCONV ACTION macros and continues with the next convention in the table.

If an ICHNCONV SELECT macro is not coded, RACF unconditionally processes the actions that are specified on the ICHNCONV ACTION macros, and continues as specified on the ICHNCONV END macro.

All ICHNCONV SELECT macros for a naming convention must follow the ICHNCONV DEFINE macro and precede any ICHNCONV ACTION macros.

COND=(condition)

Specifies the conditions that must exist before the naming convention processes the data set name.

The "condition" may be a simple comparison condition of the form:

```
COND=(variable,operator,operand)
```

You can also use a "compound condition" that is formed by linking two or more ICHNCONV SELECT macros with logical AND and OR operators:

```
COND=(variable,operator,operand,AND)
```

or

```
COND=(variable,operator,operand,OR)
```

If a naming convention contains more than one ICHNCONV SELECT macro, all of the SELECT macros except the last must contain either AND or OR to link it to the following macro. The last (or only) ICHNCONV SELECT cannot have AND or OR specified. RACF evaluates compound conditions in the order specified. AND and OR have equal precedence. Each operation is performed in order, with no "short-circuit" evaluation. The final result of a compound condition is always:

```
(...(((conv1 op1 conv2) op2 conv3) op3 conv4)...opn convn+1) ...
```

where any "op" can be either AND or OR.

variable

Specifies the variables that the convention can reference. Valid variables are:

GQ

Input qualifiers

G

Input qualifier array subscript

UQ

Output qualifiers

U

Output qualifier array subscript

QUAL

Character qualifier

QCT

Initial number of qualifiers

NAMETYPE

Type of data set

EVENT

Event code

VOLUME

Volume serial numbers

V

Volume serial number array subscript

VCT

Number of volumes

OLDVOL

Volume serial of old volume

WKX

Temporary work variable

WKY

Temporary work variable

WKZ

Temporary work variable

WKA

Temporary work variable

WKB

Temporary work variable

WKC

Temporary work variable

RACUID

Caller's user ID

RACGPID

Caller's current connect group

RACUID3

User ID for third-party RACHECK

RACGPID3

Group used for the third-party RACHECK

RACF initializes the variables before the first convention. ICHNCONV passes any changes to a variable to subsequent conventions, but only changes made to the variables UQ, QUAL, and NAMETYPE are passed back to the RACF module that called the naming convention table processing module.

You can reference character and hexadecimal variables by substring; for example, (variable, subscript, substring-start, substring-end). If the variable does not accept subscripts or you omit the subscript, you must code a comma to show that the subscript is omitted. Variables cannot be used to define the extents of substrings. For example, (GQ,2,1,3) refers to the first three characters of the second input qualifier; (EVENT,,2,2) refers to the second byte of the event code.

Example: The definition of the data set BOB.SAMPLE.DATASET on volume 111111, when the naming convention table processing module was called during a TSO session when user RACUSR1 was connected to group RACGRP1, would lead to the following set of initial variables:

```
(GQ,1) = BOB
(GQ,2) = SAMPLE
(GQ,3) = DATASET
(GQ,4) to (GQ,22) = blank
(UQ,0) = blank
(UQ,1) = BOB
(UQ,2) = SAMPLE
(UQ,3) = DATASET
(UQ,4) to (UQ,22) = blank
QCT = 3
QUAL = BOB
NAMETYPE = UNKNOWN
EVENT = X'0201'
(VOLUME,1) = 111111
VCT = 1
G, U, V = -1
WKX, WKY, WKZ = 0
WKA, WKB, WKC = blank
OLDVOL = blank
RACUID = RACUSR1
RACUID3 = blank
RACGPID = RACGRP1
RACGPID3 = blank
```

GQ

input qualifiers of the data set name.

G

input qualifier array subscript.

UQ

output qualifiers of the data set name.

U

output qualifier array subscript.

GQ and UQ are arrays containing the qualifiers of the data set name with the high-level qualifier of the name in (GQ,1) and (UQ,1). G and U are halfword variables that are used to hold subscripts to the GQ and UQ arrays. G and U are initialized to negative one (-1), which is out of the range of valid subscripts.

Initially the input and output qualifiers are identical; but if the contents of the output qualifiers are changed, the new contents are used as the new data set name. Each qualifier is a 44-byte character field padded on the right with blanks. (The field does not include the periods that separate qualifiers.) Initially, (UQ,0) is blank and is reserved for the convention to set as the new high-level qualifier.

If the name produced by the naming conventions table is longer than 44 characters, it is truncated to 44 characters. Thus, the highest possible number of qualifiers (or the highest possible value for the subscripts G and U) is 22.

Naming conventions should ensure that none of the UQ fields contain anything after the 8th position, because RACF follows the MVS JCL rules for data set names that are not enclosed within quotations, and requires that qualifiers be at most 8 characters in length.

If you use GQ or UQ in an ICHNCONV SELECT macro without a subscript, RACF tests the condition for each qualifier in turn until the condition is true. The variables G and U are set to the subscript of the qualifier for which the condition was true, G for the conditions using GQ and U for those involving UQ. If the condition is not true, the subscript variable is negative one (-1).

For all conditions except NE (not equal), the implied linkage is OR; for the NE condition, the implied linkage is AND. For example,

```
SELECT COND=(GQ,EQ,'ABC')    means
```

```
SELECT COND=((GQ,0),EQ,'ABC',OR)
SELECT COND=((GQ,1),EQ,'ABC',OR) ...
```

while

```
SELECT COND=(GQ,NE,'ABC')    means
```

```
SELECT COND=((GQ,0),NE,'ABC',AND)
SELECT COND=((GQ,1),NE,'ABC',AND) ...
```

You can use any numeric variable as a subscript for GQ or UQ. If RACF encounters an out-of-range subscript (for example, -1 or 23), RACF uses blanks for the comparison.

If GQ or UQ is in an ICHNCONV ACTION macro without a subscript, RACF uses the current value of G or U, in that order, as the subscript.

QUAL

An 8-byte character qualifier that RACF uses in authority checking to determine if the data set is the user's data set or a group data set.

QUAL is initially the data set high-level qualifier. If the high-level qualifier is not a user ID or group name, you should set QUAL to a user ID or group name. Setting QUAL, however, is not the same as setting the data set high-level qualifier. QUAL and the high-level qualifier are two separate fields, used for different RACF processing. Therefore, if you change QUAL, you probably want to set (UQ,0) to the same value as QUAL, especially for generic profile names.

QCT

A 2-byte binary field containing the initial number of qualifiers in the data set name.

NAMETYPE

Indicates whether the data set is a user or group data set.

NAMETYPE initially has the value UNKNOWN but a convention action may set the value to be USER or GROUP. The three special constant values UNKNOWN, USER, and GROUP may be used to test and set the value of this field. NAMETYPE is available only when the caller is RACDEF.

If the convention sets the value to USER or GROUP, RACF ensures that an appropriate user or group exists and fails the RACF or ADDSD if not.

EVENT

A 2-byte hexadecimal field containing the event code that is currently passed to the exit routine.

Values that EVENT can have are:

```
X'0100' - RACHECK (see note 1)
X'0201' - RACDEF DEFINE (RENAME new name)
X'0202' - RACDEF RENAME (OLD name)
X'0203' - RACDEF ADDVOL
X'0204' - RACDEF DELETE
X'0205' - RACDEF CHGVOL
X'0301' - ADDSD SET
X'0302' - ADDSD NOSET
X'0303' - ADDSD MODEL
X'0401' - ALTDSD SET
X'0402' - ALTDSD NOSET
X'0501' - DELDSD SET
X'0502' - DELDSD NOSET
X'0601' - LISTDSD prelocate (see note 2)
```

X'0602' - LISTDSO DATASET postlocate (see note 2)
 X'0603' - LISTDSO ID or PREFIX postlocate (see note 2)
 X'0701' - PERMIT TO-resource
 X'0702' - PERMIT FROM-resource
 X'0801' - SEARCH prelocate (see note 2)
 X'0802' - SEARCH postlocate (see note 2)
 X'0900' - IRRUT100 postlocate (see note 2)
 X'0D00' - RACXTRT

Notes:

1. RACHECK may be called by the RACROUTE interface or internally by RACF commands such as LISTDSO and SEARCH. If RACHECK is invoked by the RACROUTE macro, the name passed to naming conventions is in external user-specified format. However, if the command processors call RACHECK, the name may be in either format. Because no indicator of the type of call being made is passed to the naming conventions table, the naming conventions table must determine if the name is in internal format and switch it to external format (or if the name is in external format it must switch it to internal format) for this event code.
2. Prelocate means before a profile is located; these events (including all of those without a note) are passed a name in the external, user-specified format.

 Postlocate means after a profile is located but before it is displayed; these events are passed a name in the internal RACF format and the naming conventions processing should include converting it back to the external, user format.
3. Error messages displayed by the command processors may use the RACF internal format of the name so the message may be used to determine the real profile name that RACF attempted to locate.

VOLUME

An array of volume serial numbers for volumes containing the data set. Each volume is a 6-byte character field.

V

A 2-byte variable that contains a subscript to the volume array. V is initialized to -1.

VOLUME is not available for generic data set profiles and is not available from commands if the VOLUME keyword was not specified. An attempt to reference nonexistent volumes (subscript 0 or greater than the number of volumes in the VOLUME array) results in a VOLUME parameter which contains *BLANK as a character string.

If you reference VOLUME in an ICHNCONV SELECT macro without a subscript, RACF tests the condition for each volume in turn until the condition is true. The variable V is set to the subscript of the volume for which the condition was true. If the condition is not true, the subscript variable is negative one (-1).

For all conditions except NE (not equal), the implied linkage is OR; for the NE condition, the implied linkage is AND. For example,

SELECT COND=(VOLUME,EQ,'ABC') means

SELECT COND=((VOLUME,1),EQ,'ABC',OR)
 SELECT COND=((VOLUME,2),EQ,'ABC',OR) ...

while

SELECT COND=(VOLUME,NE,'ABC') means

SELECT COND=((VOLUME,1),NE,'ABC',AND)
 SELECT COND=((VOLUME,2),NE,'ABC',AND) ...

You may use any numeric variable as a subscript for VOLUME. If VOLUME is in an ICHNCONV ACTION macro without a subscript, RACF uses the current value of V as the subscript.

VCT

A 2-byte binary field containing the number of volumes in the VOLUME array. If volume information is not available, VCT has a value of zero.

OLDVOL

A 6-byte character field containing the volume serial number of the volume that the data set currently resides on. This field is available during a RACDEF ADDVOL or RACDEF CHGVOL request.

WKX, WKY, WKZ

These are 2-byte binary fields that may be used as temporary work variables to save subscripts and other numeric data within and between conventions.

WKA, WKB, WKC

These are 8-byte character fields that may be used as temporary work variables to save qualifiers and other non-numeric data within and between conventions.

RACUID

The caller's user ID.

Note: If naming convention table processing is invoked from an environment where no ACEE is present to define the current user, a default value of * is used for RACUID.

RACGPID

The caller's current connect group.

Note: If naming convention table processing is invoked from an environment where no ACEE is present to define the current user, a default value of * is used for RACGPID.

RACUID3

The user ID used for third-party RACHECK.

RACGPID3

The group that is used for third-party RACHECK

Note: For RACROUTE REQUEST=AUTH (event code X'0100'):

- RACUID and RACGPID are the user and group that is used for non-third-party authorization checking. They are acquired from an address space level ACEE, a task level ACEE, or the ACEE= parameter on the RACROUTE REQUEST macro.
- RACUID3 and RACGPID3 are used for third-party authorization checking. They contain blanks for all other event codes. If the REQUEST=AUTH specifies a GROUP and no USERID, RACUID3 is *NONE*. If USERID is specified and no GROUP, RACGPID3 is blanks. RACUID3 and RACGPID3 might not contain a valid user or group. Their values are obtained from the RACROUTE REQUEST=AUTH parameters and have not been validated by RACF at the time the naming convention was called.

operator

Specifies the conditional operator: Valid operators are:

EQ

Equal

GT

Greater than

LT

Less than

GE

Greater than or equal

LE

Less than or equal

NE

Not equal

operand

Specifies a variable, a literal, or one of the following special symbols for use with the NAMETYPE variable:

- USER
- GROUP
- UNKNOWN

The operand used should match the length and type of the variable. If the length does not match, RACF performs padding or truncation in the normal manner. If the type does not match, the results are unpredictable.

If operand is specified as a literal, it can be:

- A character string enclosed in quotations
- A decimal number
- A hexadecimal string in the form X'string'

ICHNCONV ACTION

An ICHNCONV ACTION macro changes the value of variables. Use these macros to convert the data set name to the standard RACF format. RACF processes the ACTION macros in sequence.

The format of the ICHNCONV ACTION macro is:

```
[label] ICHNCONV ACTION,SET=(variable,value)
```

ACTION

Identifies a naming convention action. You can code multiple ICHNCONV ACTION macros.

SET=(variable,value)

Changes the qualifiers of a data set name and other variables.

variable

Specifies the variables that the convention can reference and set. See the preceding description of ICHNCONV SELECT for a description of these variables.

The following variables can be set:

- UQ
- QUAL
- G
- U
- V
- NAMETYPE
- WKA, WKB, WKC
- WKX, WKY, WKZ

value

Specifies the value given to the variable. Value can be another variable, a literal, or one of the following special symbols for use with the NAMETYPE variable:

- USER
- GROUP
- UNKNOWN

The value assigned to a variable should match the length and type of the variable. If the length does not match, RACF performs padding or truncation in the normal manner. If the type does not match, the results are unpredictable.

If you specify value as a variable, it can be any of the variables defined in the description of ICHNCONV SELECT.

If value is a literal, it can be:

- A character string enclosed in quotation marks
- A decimal number
- A hexadecimal string of the form X'string'

ICHNCONV END

An ICHNCONV END macro terminates the naming convention.

The format of the ICHNCONV END macro is:

```
[label] ICHNCONV  END,NEXT=(convention name|'SUCCESS'|'NEXT'|'ERROR')
```

END

Identifies the end of the naming convention. Each convention must have one ICHNCONV END.

NEXT= (convention name|'SUCCESS'|'NEXT'|'ERROR')

Specifies where control goes after this convention executes, if the conditions specified in the ICHNCONV SELECT macros have been met or if there are no ICHNCONV SELECT macros in this convention.

If NEXT=convention name, processing continues with the specified convention and skips intervening conventions in the table. The specified convention must not precede the current convention in the table; otherwise, the RACF request fails.

If NEXT='NEXT', processing continues with the next convention in sequence. If NEXT='NEXT' is coded or defaulted to on the last convention in the table, processing is the same as if NEXT='SUCCESS' was coded.

If NEXT='SUCCESS', then the convention processing routine bypasses further convention processing and returns "a successful name processing" return code to the RACF routine that called it. The RACF routine continues to process normally using the name returned by the convention processing routine.

If NEXT='ERROR', then the convention processing routine bypasses further processing and returns "an invalid data set name" return code to the RACF routine that called it. The RACF routine terminates processing and fail the request.

ICHNCONV FINAL

An ICHNCONV FINAL macro terminates the naming convention table. (The naming convention table has one ICHNCONV FINAL macro.)

The format of the ICHNCONV FINAL macro is:

```
[label] ICHNCONV  FINAL
```

FINAL

Identifies the end of the naming conventions table. There must be only one ICHNCONV FINAL macro in the table and it must be the last entry in the table.

Example of a naming convention table

The following example of a naming convention table illustrates some ways that a table could be coded.

The first convention checks for data sets that are already in the correct RACF format, with a user ID or group name in the high-level qualifier or system data sets that start with the characters SYS. This convention bypasses all further checks because no further changes are needed.


```

ICHNCONV DEFINE,NAME=CHECK1
ICHNCONV SELECT,COND=((GQ,1),EQ,RACUID,OR)
ICHNCONV SELECT,COND=((GQ,1),EQ,RACGPID,OR)
ICHNCONV SELECT,COND=((GQ,1,1,3),EQ,'SYS')
ICHNCONV END,NEXT='SUCCESS'

```

This convention checks for data set names that have three or more qualifiers and any qualifier is the user's ID. The user ID is moved to the start of the name and deleted from its current position. ICHNCONV sets the type indicator and processing continues with convention CHECK4.

```

ICHNCONV DEFINE,NAME=CHECK2
ICHNCONV SELECT,COND=(QCT,GE,3,AND)
ICHNCONV SELECT,COND=(GQ,EQ,RACUID)
ICHNCONV ACTION,SET=(NAMETYPE,USER)
ICHNCONV ACTION,SET=((UQ,0),(GQ,G))
ICHNCONV ACTION,SET=((UQ,G),' ')
ICHNCONV END,NEXT=CHECK4

```

For all data sets that did not pass the first two conventions the first four characters of the third and fourth qualifiers are concatenated to form a new fifth qualifier. The user's current connect group becomes a high-level qualifier. Processing continues (by default) with the next convention.

```

ICHNCONV DEFINE,NAME=CHECK3
ICHNCONV ACTION,SET=((UQ,0),RACGPID)
ICHNCONV ACTION,SET=((UQ,5,1,4),(GQ,3,1,4))
ICHNCONV ACTION,SET=((UQ,5,5,8),(GQ,4,1,4))
ICHNCONV ACTION,SET=(NAMETYPE,GROUP)
ICHNCONV END

```

The installation has decided to enforce a standard that all three-qualifier data set names must have a data set type code as the last qualifier. Any qualifiers that are not in the list will cause the name to be rejected.

```

ICHNCONV DEFINE,NAME=CHECK4
ICHNCONV SELECT,COND=(QCT,EQ,3,AND)
ICHNCONV SELECT,COND=((GQ,3),NE,'PLI',AND)
ICHNCONV SELECT,COND=((GQ,3),NE,'DATA',AND)
ICHNCONV SELECT,COND=((GQ,3),NE,'COBOL',AND)
ICHNCONV SELECT,COND=((GQ,3),NE,'ASM')
ICHNCONV END,NEXT='ERROR'

```

The ICHNCONV FINAL macro terminates the table. An assembly language END statement is necessary to terminate the assembly.

```

ICHNCONV FINAL
END

```

ICHRFRTB macro

The RACF router table is an optional table that allows an installation to bypass RACF processing for a class or for a requester and subsystem combination. An entry is required in this table for a class only if the class does not require RACF to be called on each invocation of the RACROUTE macro. The same is true for each class, requester, and subsystem combination; an entry is only required if RACF is not to be called. All entries that specify ACTION=RACF are optional. The RACF router treats each class and combination of requester and subsystem that does not have an entry in the router table as if it has an entry in the table specifying ACTION=RACF.

An installation can use the RACF router table to change the processing for a class that IBM supplies. For example, if a tape library product wants to bypass some DFP-issued OCEOV calls, the installation can create a RACF router table with one or more entries that specify ACTION=NONE for combinations of class, requester, and the OCEOV subsystem.

Example: An installation could use the following macro invocations to create a router table that bypasses OCEOV calls.

```
ICHRFRTB CLASS=DATASET,REQSTOR=CLOSE,SUBSYS=OCEOV,ACTION=NONE
ICHRFRTB CLASS=DATASET,REQSTOR=TAPEOPEN,SUBSYS=OCEOV,ACTION=NONE
ICHRFRTB CLASS=TAPEVOL,REQSTOR=TAPEOPEN,SUBSYS=OCEOV,ACTION=NONE
ICHRFRTB CLASS=DATASET,REQSTOR=TAPEEOV,SUBSYS=OCEOV,ACTION=NONE,
ICHRFRTB CLASS=TAPEVOL,REQSTOR=CLOSE,SUBSYS=OCEOV,ACTION=NONE
ICHRFRTB TYPE=END
```

The ICHRFRTB macro generates entries in the RACF router table, module ICHRFRTB01.

The ICHRFRTB macro definition is as follows:

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

ACTION=

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

NONE

Specifies that no action is to be taken for this entry.

RACF

Specifies that RACF is to be called for this entry.

CLASS=class name

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

REQSTOR=requestor-name

Specifies the 8-character requester name. Installations should begin requester names with a # (X'7B'), @ (X'7C') or \$ (X'5B'), because requester names supplied by IBM do not begin with those characters. If you do not specify a requester name, the default is a string of 8 blanks. If you code REQSTOR, you should also code the CLASS operand.

SUBSYS=subsystem-name

Specifies the 8-character subsystem name. Installations should begin subsystem names with a # (X'7B'), @ (X'7C') or \$ (X'5B'), because subsystem names supplied by IBM do not begin with such characters. If no subsystem name is specified, it defaults to a string of 8 blanks. This operand should not be coded unless CLASS is also specified.

TYPE=END

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

Chapter 2. Panel driver interface module (ICHSPF03)

Installations can implement a panel driver interface between an application program and the RACF panels. In order to use the panel driver interface, the programmer who writes the interface should be familiar with TSO CLIST or ISPF programming techniques.

Invoking the panel driver interface

When you invoke the panel driver interface module (ICHSPF03), your program must pass it the following three parameters as ISPF variables:

ICHFUNCT

the type of function that ICHSPF03 is to perform: you may specify blank, ADD, CHG, DEL, ACC, and DSP.

ICHRESCL

the name of the resource class: for example; group, user, data set, or any of the general resource classes defined to RACF in the class descriptor table (CDT). The classes supplied by IBM are in [“Supplied class descriptor table entries”](#) on page 523. Other classes can be added by your installation.

ICHRESNM

a resource name within the resource class, supplied with the resource class by the programmer writing the interface.

These parameters are passed to ICHSPF03 using the ISPLINK SELECT service and the function variable pool.

ICHSPF03 matches the first two arguments that are passed in the parameter list (function and resource class) to the panel mapping table to determine which RACF panel to display.

Panel mapping table

Function	Resource class	Panel ID
bbb	bbbbbbbbb	ICHP00
bbb	DATASET	ICHP10
ADD	DATASET	ICHP11
CHG	DATASET	ICHP12
DEL	DATASET	ICHP13
ACC	DATASET	ICHP14
DSP	DATASET	ICHP18
bbb	general	ICHP20
ADD	general	ICHP21
CHG	general	ICHP22
DEL	general	ICHP23
ACC	general	ICHP24
DSP	general	ICHP28
bbb	GROUP	ICHP30

Function	Resource class	Panel ID
ADD	GROUP	ICHP31
CHG	GROUP	ICHP32
DEL	GROUP	ICHP33
DSP	GROUP	ICHP38
bbb	USERID	ICHP40
ADD	USERID	ICHP41
CHG	USERID	ICHP42
DEL	USERID	ICHP43
DSP	USERID	ICHP48

Notes:

1. In the table above, 'general' stands for any valid general resource class.
2. bbb... represents blanks.

If the caller enters a *resource* class that is not defined in the table, the argument defaults to general. If the caller enters a *function* that is not defined in the table, ICHSPF03 issues an error message.

ICHSPF03 issues a VREPLACE within the panel driver interface to update the shared variable pool with the parameters passed from the user's function panel. ICHSPF03 then places those parameters in the panels where variables are required to identify the resource. Thus, the user does not have to constantly retype parameters. If there is an error in the caller's parameter list, ICHSPF03 issues an error message.

The ISPLINK call

The following is an example of a declare for the RACF panel driver interface:

```
DCL Buffer Char(13) Init('PGM(ICHSPF03)');
```

The following is the format of a call to the RACF panel driver interface:

```
CALL ISPLINK(SELECT, LENGTH(BUFFER), BUFFER)
```

ICHSPF03 issues an ISPLINK call for the target RACF panel. Upon return from the RACF panel after performing the requested function, RACF enters a return code in Register 15.

The return code is one of the following:

- 0 = successful completion of the function
- 12 = invalid function code specified
- 16 = variable not defined in the function pool

The way in which ICHSPF03 invokes the RACF panels depends on the way the programmer coded the interface. For example, if the parameters passed were ICHFUNCT = bbb, ICHRESCL = DATASET, and ICHRESNM = A.B.C, then ICHP10 is displayed as:

```
PROFILE NAME === > A.B.C
```

Note: On this screen, you can modify the profile name.

If the parameters passed are ICHFUNCT = CHG, ICHRESCL = DASDVOL, and ICHRESNM = DPT67V, then ICHP22 are displayed as:

```
CLASS: DASDVOLPROFILE NAME: DPT67V
```

Note: In this case, you cannot modify the class or the profile name.

Example of a RACF panel interface coding sequence

The following is an example of a user-written coding sequence to create an interface to the RACF panels. You can also call the interface from a CLIST in a similar fashion.

```
*Create Variables in the Function Variable Pool*/
Call ISPLINK(VDEFINE,'ICHFUNCT',ICHFUNCT,'CHAR',LENGTH(ICHFUNCT));
Call ISPLINK(VDEFINE,'ICHRESCL',ICHRESCL,'CHAR',LENGTH(ICHRESCL));
Call ISPLINK(VDEFINE,'ICHRESNM',ICHRESNM,'CHAR',LENGTH(ICHRESNM));
/*Copy variables from Function Pool to Shared Variable Pool*/
Call ISPLINK('VPUT','ICHFUNCT');
Call ISPLINK('VPUT','ICHRESCL');
Call ISPLINK('VPUT','ICHRESNM');
/*Call the Panel Driver Interface*/
Call ISPLINK(SELECT,LENGTH(BUFFER),BUFFER);
/*Test return code from the PDI for possible errors */
```


Chapter 3. Profile name list service routine (IRRPNL00)

RACF provides installations with a profile name list service routine (IRRPNL00) that allows TSO or other programs to call RACF to retrieve the names of profiles within a class that a given user ID can access at READ level or higher.

To perform this function, IRRPNL00 searches the RACF class descriptor table (CDT) for the class name. If the class is found and the class is processed by SETROPTS RACLIST, IRRPNL00 checks each profile name processed by SETROPTS RACLIST to see if the specified user ID is authorized to access the profile at READ level or higher. When IRRPNL00 finds a match, it places the profile name into the input work area.

IRRPNL00 begins its search with the first profile and continues its search until it checks all the profiles or until the size of the list exceeds the size of the work area.

IRRPNL00 resides in LPA. RACF loads the address of IRRPNL00 into RCVTPNL0 during RACF initialization. The caller of IRRPNL00 can use the address in RCVTPNL0.

Notes:

1. To use the profile name list service routine, you must ensure that a SETROPTS RACLIST is issued for each class name you intend to search.
2. Your program is responsible for obtaining and releasing the storage which IRRPNL00 uses to store the profile name list.
3. Callers of IRRPNL00 must be authorized in one of the following ways:
 - APF-authorized
 - Supervisor state
 - System key 0.

Invoking the profile name list service routine

When you invoke the profile name list service routine (IRRPNL00), your program must pass it the following four parameters:

Classname

an 8-character class name from which RACF derives the profile names to which the user ID has authorization

Work area length

a fullword that contains the length of the area in which IRRPNL00 is going to build the profile name list

Work Area

a fullword pointer that contains the address of the work area where IRRPNL00 is going to build the profile name list

ACEE pointer

a fullword pointer that contains the address of the ACEE for the user ID for whom profile authorization is being determined

Note: If the ACEE pointer is zero, IRRPNL00 uses the ACEE pointed to by TCBSENV.

If the TCBSENV pointer is zero, IRRPNL00 attempts to use the ASXBSENV field.

If the ASXBSENV field is zero, IRRPNL00 returns an error return code and reason code.

The calling program passes these parameters to IRRPNL00 using the CALL command.

If IRRPNL00 is being called by a RACF exit, it must be invoked using the SYNCH macro. See [z/OS MVS Programming: Assembler Services Guide](#).

Format of returned profile name list

A FIXED(31) count field which precedes the profile name list contains the total count of profile names returned by IRRPNL00. The format of the profile name list when it is placed in the work area is as follows:

NAME LENGTH

A 2-byte length of the profile name

FLAGS

A 1-byte flag field (only first bit used)

PROFILE NAME

A variable-length profile name

Note: The first bit of the flag field byte is on if the profile is a generic profile.

Return codes

The return codes from IRRPNL00 follow RACF conventions with a return code of 0 indicating a successful search. The return codes are as follows:

Note: All return and reason codes are shown in hexadecimal.

The following return codes are returned in register 15, and the reason codes in register 0.

Return Code

Meaning

00

The profile name list function completed successfully.

04

No profiles found for which user ID had at least read access.

08

No profile entries found for that class. Indicates that either no profiles existed for the input class or the input class was not processed by SETROPTS RACLIST.

0C

The work area was not large enough to hold all the profile names.

14

Profile name list parameter error

Reason Code

Meaning

04

No ACEE available.

08

Work area too small to contain a single profile.

10

Input class name not valid.

18

Unable to establish ESTAE environment.

1C

Nonzero return code from the data space search routine: Return Code and Reason Codes are returned in the low-order and high-order halfwords of Register 0. Most are internal RACF codes indicating an error in RACF, except for the following:

Reason Code

Meaning

00080008

Class not processed by SETROPTS RACLIST

00080018

RACLIST data space could not be accessed due to an ALESERV failure.

20

ACEE has a default UTOKEN

24

The ACEE UTOKEN has a port-of-entry class indicated but no port-of-entry name is supplied.

Chapter 4. Date conversion routine

RACF provides installations with the IRRDCR00 module, which is the date conversion routine that enables programs to specify and identify dates beyond the end of the 20th century.

Using this routine, installation and vendor applications can call RACF to convert a three-byte packed-decimal date to a four-byte packed-decimal date. The three-byte date has the form *yydddF*, and the four-byte date has the form *ccyydddF*, where *cc* is 00 for years 1971 through 1999 and is 01 for years 2000 through 2070. In the three-byte form, the routine interprets the year as 19yy when yy is 71 or higher and as 20yy when yy is less than 71.

This routine resides in the LPA. The system loads the address of this routine in RCVTDATP and sets bit RCVTD4OK (X'08') in flag byte RCVTMFLG during RACF initialization. The routine's caller can use the address in RCVTDATP.

Invoking the date conversion routine

When a program invokes the date conversion routine, the program must pass two parameters to it:

Three-byte date

a three-byte field containing a packed-decimal date (format *yydddF*)

Four-byte date

a four-byte field

Note: This routine runs in the caller's mode, state, and key. Recovery is handled by the calling program.

Format of returned converted date

The routine returns a four-byte packed-decimal date whose format is either *00yydddF* (for 1971-1999) or *01yydddF* (for 2000-2070).

If *ddd* is 000 in the three-byte *yydddF* date field passed to the routine, the routine returns 00 for *cc* (indicating a year 19yy), regardless of what *yy* is.

Return code

The return code from this routine follows RACF conventions with a return code of 00 (X'00') indicating successful date conversion. This code is sent to register 15.

The return code is as follows:

Note: The return code is shown in hexadecimal.

Return Code	Meaning
-------------	---------

00	The date conversion function completed successfully.
----	--

Chapter 5. SMF records

RACF produces three SMF records:

Type 80

Produced during RACF processing

Type 81

Produced at the completion of RACF initialization and the SETROPTS command

Type 83

Produced during RACF and z/OS component processing

The type 83 record is generated under the following circumstances: SETROPTS MLACTIVE is in effect and a RACF command (ADDSD, ALTDSD, DELDSD) has been issued that changed the security label of a data set profile.

Security events detected by non-RACF components.

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

For sorting purposes, the RACF report writer reformats SMF records (types 20, 30, 80, 81 and 83) and uses these reformatted records as input to the modules that produce the RACF reports. There are two types of reformatted records: reformatted process records and reformatted status records. If you want to use the RACF report writer exit (ICHRSMFE) to produce additional reports or to add additional record selection criteria, you should familiarize yourself with the layouts of these reformatted records.

Table 1 on page 31 shows each RACROUTE request type and the corresponding independent system macro.

Guideline: Use a RACROUTE request rather than an independent system macro.

Table 1. RACROUTE request types and corresponding independent RACF system macro. Shows RACROUTE request types and corresponding independent RACF system macros	
RACROUTE request type	Independent RACF system macro
REQUEST=AUTH	RACHECK
REQUEST=DEFINE	RACDEF
REQUEST=EXTRACT	RACXTRT
REQUEST=FASTAUTH	FRACHECK
REQUEST=LIST	RACLIST
REQUEST=STAT	RACSTAT
REQUEST=VERIFY	RACINIT
REQUEST=VERIFYX	RACINIT

Record type 80: RACF processing record

RACF writes record type 80 for the following detected events:

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

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

- **Authorized attempts to enter the system.** RACF provides a RACROUTE REQUEST=VERIFY option to log successful signons and signoffs including ENVIR=CREATE or ENVIR=DELETE signons and signoffs. For the LOG keyword on the RACROUTE REQUEST=VERIFY macros, LOG=ALL or LOG=ASIS may be specified to control the generation of log records for RACROUTE REQUEST=VERIFY. The value of the LOG keyword is passed to both the RACROUTE REQUEST=VERIFY preprocessing and postprocessing installation exits. Both exits are invoked before the generation of a log record, and the LOG keyword value can be changed for both exits.
- **Authorized accesses or unauthorized attempts to access RACF-protected resources.** During RACF processing of a RACROUTE REQUEST=AUTH or REQUEST=DEFINE macro instruction, RACF found that one of the following events occurred:
 1. The user was permitted access to a RACF-protected resource and allowed to perform the requested operation.
 2. The user did not have sufficient access or group authority to access a RACF-protected resource, or supplied invalid data while attempting to perform an operation on a RACF-protected resource.

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

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

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

For the preceding events, a RACROUTE REQUEST=AUTH exit routine can modify the logging options by changing the LOG parameter on a RACROUTE REQUEST=AUTH macro instruction from ASIS to NOFAIL, NONE, or NOSTAT, or by unconditionally requesting or suppressing logging with the logging control field. For information about the LOG parameter of a RACROUTE REQUEST=AUTH macro instruction, see *z/OS Security Server RACROUTE Macro Reference*. For information about the logging options of the ADDSD, ALTDSD, ALTUSER, RALTER, RDEFINE, and SETROPTS commands, see *z/OS Security Server RACF Command Language Reference*.

- **Authorized or unauthorized attempts to modify profiles on a RACF database.** During RACF command processing, RACF found that a user with the AUDITOR attribute specified that the following be logged:
 1. All detected changes to a RACF database by RACF commands or a RACROUTE REQUEST=DEFINE
 2. All RACF commands (except LISTDSD, LISTGRP, LISTUSER, RLIST, and SEARCH) issued by users with the SPECIAL attribute
 3. All violations detected by RACF commands (except LISTGRP, LISTUSER, RLIST, and SEARCH)
 4. Every RACROUTE REQUEST=AUTH and RACROUTE REQUEST=DEFINE issued for the user and all RACF commands (except LISTGRP, LISTUSER, RLIST and SEARCH) issued by the user

In the first three cases, RACF writes records if a user with the AUDITOR attribute specified AUDIT, SAUDIT, and CMDVIOL, in that order, on the SETROPTS command. In the fourth case, RACF writes the records if a user with the AUDITOR attribute specified UAUDIT on the ALTUSER command.

- **Generation or evaluation of a PassTicket via the RCVTPTGN, R_ticketserv or R_GenSec services.** PassTicket use during normal logon is reflected in the SMF type 80 record generated for an authorized or an unauthorized attempt to enter the system (see above), and does not result in a separate SMF record.

You can use SMF records to:

- Track the total use of a sensitive resource (if the ALL option is set)

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

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

Format of SMF type 80 records

SMF type 80 records contain the following information:

- The record type
- Time stamp (time and date)
- Processor identification
- Event code and qualifier (explained in [“Table of event codes and event code qualifiers” on page 41](#))
- User identification
- Group name
- A count of the relocate sections
- Authorities used to successfully execute commands or access resources
- Reasons for logging

Note: In general, RACF searches for reasons for auditing an event until it finds one, then audits without looking for more reasons that might also have caused auditing. This means that most RACF SMF records will show only one reason for auditing, even though several might apply (and in a few cases, more than one might actually be shown in the record). There are many places in RACF that audit, and the order of checking is not the same in all places, so the audit reason that will be used is not entirely predictable. In some cases it would not even be possible for RACF to look for additional potential audit reasons without causing adverse performance impact to the system. For example, SPECIAL users are often granted access to a resource without even reading the resource profile that protects it, so no information is available about what auditing options the profile might have requested.

- Command processing error flag
- Foreground user terminal ID
- Foreground user terminal level number
- Job log number (job name, entry time, and date)
- RACF version, release, and modification number
- Security label of user

The data in the relocate sections is explained in the following tables:

- [“Table of relocate section variable data” on page 56](#)
- [“Table of extended-length relocate section variable data” on page 65](#)
- [“Table of data type 6 command-related data” on page 81](#)

The log record RACF creates is a standard SMF record with the type 80 format. [Table 2 on page 34](#) describes the format of the type 80 record.

Table 2. Format of the SMF type 80 record

Offset (Dec.)	Offset (Hex)	Name	Length	Format	Description
0	0	SMF80LEN	2	Binary	Record length.
2	2	SMF80SEG	2	Binary	Segment descriptor.
4	4	SMF80FLG	1	Binary	<p>System indicator</p> <p>Bit</p> <p>Meaning when set</p> <p>0-2 Reserved for IBM's use.</p> <p>3 MVS/ or 5</p> <p>4 MVS/</p> <p>5 MVS/</p> <p>6 VS2</p> <p>7 Reserved for IBM's use.</p> <p>Note: For MVS/, bits 3, 4, 5, and 6 are on.</p>
5	5	SMF80RTY	1	Binary	Record type: 80 (X'50').
6	6	SMF80TME	4	Binary	Time of day, in hundredths of a second, that the record was moved to the SMF buffer.
10	A	SMF80DTE	4	packed	Date that the record was moved to the SMF buffer, in the form 0cyydddF (where F is the sign).
14	E	SMF80SID	4	EBCDIC	System identification (from the SID parameter).
18	12	SMF80DES	2	Binary	<p>Descriptor flags</p> <p>Bit</p> <p>Meaning when set</p> <p>0 The event is a violation.</p> <p>1 User is not defined to RACF.</p> <p>2 Record contains a version indicator (see SMF80VER).</p> <p>3 The event is a warning.</p> <p>4 Record contains a version, release, and modification level number (see SMF80VRM).</p> <p>5-15 Reserved for IBM's use.</p>
20	14	SMF80EVT	1	Binary	Event code.
21	15	SMF80EVQ	1	Binary	Event code qualifier.
22	16	SMF80USR	8	EBCDIC	Identifier of the user associated with this event (jobname is used if the user is not defined to RACF).
30	1E	SMF80GRP	8	EBCDIC	Group to which the user was connected (stepname is used if the user is not defined to RACF).
38	26	SMF80REL	2	Binary	Offset to the first relocate section from SMF80FLG.
40	28	SMF80CNT	2	Binary	Count of the number of relocate sections.

Table 2. Format of the SMF type 80 record (continued)

Offset (Dec.)	Offset (Hex)	Name	Length	Format	Description
42	2A	SMF80ATH	1	Binary	<p>Authorities used for processing commands or accessing resources. (See Note “1” on page 40.)</p> <p>Bit</p> <p>Meaning when set</p> <p>0 Normal authority check (resource access).</p> <p>1 SPECIAL attribute (command processing).</p> <p>2 ROAUDIT attribute (command processing). OPERATIONS attribute (resource access, command processing).</p> <p>3 AUDITOR attribute (command processing).</p> <p>4 Installation exit processing (resource access).</p> <p>5 Failsoft processing (resource access).</p> <p>6 Bypassed-user ID = *BYPASS* (resource access).</p> <p>7 Trusted attribute (resource access).</p>

Table 2. Format of the SMF type 80 record (continued)

Offset (Dec.)	Offset (Hex)	Name	Length	Format	Description
43	2B	SMF80REA	1	Binary	<p>Reason for logging. These flags indicate the reason RACF produced the SMF record. (See Note “2” on page 40.)</p> <p>Bit</p> <p>Meaning when set</p> <p>0</p> <p>Changes to this class of profile are being audited due to SETROPTS AUDIT(class). For Event Code 1, the class is USER, and this bit indicates that a password or password phrase was changed during logon.</p> <p>1</p> <p>User being audited.</p> <p>2</p> <p>SPECIAL or OPERATIONS user being audited. (See Note “2” on page 40.)</p> <p>3</p> <p>Access to the resource is being audited due to any of the following reasons:</p> <ul style="list-style-type: none"> AUDIT option was specified when a profile was created or changed by a RACF command User audit bits are set for UNIX files and directories, which is done by using the chaudit command Logging request was issued from the RACROUTE REQUEST=AUTH exit routine Operator granted access during failsoft processing. <p>4</p> <p>RACROUTE REQUEST=VERIFY or initACEE failure.</p> <p>5</p> <p>This command is always audited.</p> <p>6</p> <p>Violation detected in command and CMDVIOL is in effect.</p> <p>7</p> <p>Access to the entity is being audited due to either of the following settings:</p> <ul style="list-style-type: none"> The GLOBALAUDIT option in a RACF profile For UNIX files and directories, the AUDITOR audit bits, which are by using the chaudit -a command.
44	2C	SMF80TLV	1	Binary	Terminal level number of foreground user (zero if not available).
45	2D	SMF80ERR	1	Binary	<p>Command processing error flag. (See Note “3” on page 40.)</p> <p>Bit</p> <p>Meaning when set</p> <p>0</p> <p>Command had error and RACF could not back out some changes</p> <p>1</p> <p>No profile updates were made because of error in RACF processing</p> <p>2-7</p> <p>Reserved for IBM's use.</p>
46	2E	SMF80TRM	8	EBCDIC	Terminal ID of foreground user (zero if not available).
54	36	SMF80JBN	8	EBCDIC	Job name. For RACROUTE REQUEST=VERIFY and REQUEST=DEFINE records for batch jobs, this field can be zero if the job name is not available at the time of the RACROUTE REQUEST=VERIFY or REQUEST=DEFINE.

Table 2. Format of the SMF type 80 record (continued)

Offset (Dec.)	Offset (Hex)	Name	Length	Format	Description
62	3E	SMF80RST	4	Binary	Time, in hundredths of a second, that the reader recognized the JOB statement for this job. For RACROUTE REQUEST=VERIFY records for batch jobs, this field can be zero.
66	42	SMF80RSD	4	packed	Date the reader recognized the JOB statement for this job, in the form 0cyydddF (where F is the sign). For RACROUTE REQUEST=VERIFY records for batch jobs, this field can be zero.
70	46	SMF80UID	8	EBCDIC	User identification field from the SMF common exit parameter area. For RACROUTE REQUEST=VERIFY records for batch jobs, this field can be zero.
78	4E	SMF80VER	1	Binary	Version indicator (8 = Version 1, Release 8 or later). As of RACF 1.8.1, SMF80VRM is used instead.
79	4F	SMF80RE2	1	Binary	<p>Additional reasons for logging</p> <p>Bit</p> <p>Meaning when set</p> <p>0</p> <p>Security level control for auditing.</p> <p>1</p> <p>VMEVENT Auditing.</p> <p>2</p> <p>Class being audited due to SETROPTS LOGOPTIONS.</p> <p>3</p> <p>Audited due to SETROPTS SECLABELAUDIT.</p> <p>4</p> <p>Entity audited due to SETROPTS COMPATMODE.</p> <p>5</p> <p>Audited due to SETROPTS APPLAUDIT.</p> <p>6</p> <p>Audited because user not defined to z/OS UNIX</p> <p>7</p> <p>Audited because user does not have appropriate authority for z/OS UNIX</p>

Table 2. Format of the SMF type 80 record (continued)

Offset (Dec.)	Offset (Hex)	Name	Length	Format	Description
80	50	SMF80VRM	4	EBCDIC	FMID for RACF 2020 RACF 2.2 and OS/390 Security Server (RACF) V1 R2 2030 OS/390 Security Server (RACF) V1 R3 2040 OS/390 Security Server (RACF) V2 R4 2060 OS/390 Security Server (RACF) V2 R6 2608 OS/390 Security Server (RACF) V2 R8 7703 OS/390 Security Server (RACF) V2 R10 and z/OS Security Server (RACF) V1 R1 7705 z/OS Security Server (RACF) V1 R2 7706 z/OS Security Server (RACF) V1 R3 7707 z/OS Security Server (RACF) V1 R4 7708 z/OS Security Server (RACF) V1 R5 7709 z/OS Security Server (RACF) V1 R6 7720 z/OS Security Server (RACF) V1 R7 7730 z/OS Security Server (RACF) V1 R8 7740 z/OS Security Server (RACF) V1 R9 7750 z/OS Security Server (RACF) V1 R10 7760 z/OS Security Server (RACF) V1 R11 7770 z/OS Security Server (RACF) V1 R12 7780 z/OS Security Server (RACF) V1 R13 7790 z/OS Security Server (RACF) V2 R1 77A0 z/OS Security Server (RACF) V2 R2 77B0 z/OS Security Server (RACF) V2 R3 77C0 z/OS Security Server (RACF) V2 R4 77D0 z/OS Security Server (RACF) V2 R5 77E0 z/OS Security Server (RACF) V3 R1 77F0 z/OS Security Server (RACF) V3 R2
84	54	SMF80SEC	8	EBCDIC	Security label of the user.
92	5C	SMF80RL2	2	Binary	Offset to extended-length relocate sections from SMF80FLG.

Table 2. Format of the SMF type 80 record (continued)

Offset (Dec.)	Offset (Hex)	Name	Length	Format	Description
94	5E	SMF80CT2	2	Binary	Count of extended-length relocate sections.
96	60	SMF80AU2	1	Binary	Authority used continued Bit Meaning when set 0 z/OS UNIX superuser (Both UID(0) and BPX.SUPERUSER) 1 z/OS UNIX system function 2-7 Reserved for IBM's use.
97	61	SMF80RSV	1	Binary	Reserved for IBM's use
Relocate section: See “Table of relocate section variable data” on page 56.					
0	0	SMF80DTP	1	Binary	Data type
1	1	SMF80DLN	1	Binary	Length of data that follows
2	2	SMF80DTA	1-255	mixed	Data
Extended-length relocate section: See “Table of extended-length relocate section variable data” on page 65.					
0	0	SMF80TP2	2	Binary	Data type
2	2	SMF80DL2	2	Binary	Length of data that follows
4	4	SMF80DA2	variable	EBCDIC	Data

Table 2. Format of the SMF type 80 record (continued)

Offset (Dec.)	Offset (Hex)	Name	Length	Format	Description
Notes:					
<p>1. SMF80ATH: These flags indicate the authority checks made for the user who requested the action. The RACF commands use bits 0, 1, and 3; the RACF requests use bits 0, 2, and 4-7.</p> <ul style="list-style-type: none"> • Bit 0 indicates that the user's authority to issue the command or SVC was determined by the checks for a user with the SPECIAL, OPERATIONS, AUDITOR, or ROAUDIT attribute. This bit indicates that the tests were made, not that the user passed the tests and has authority to issue the command. This bit is not set on if the user has the AUDITOR attribute and entered the command with only those operands that require the AUDITOR attribute. • Bit 1 indicates that the user has the SPECIAL attribute and used this authority to issue the command. If the user also has the AUDITOR or ROAUDIT attribute and entered the command with only those operands that require the AUDITOR or ROAUDIT attribute, this bit is not set on because the user did not use their authority as a user with the SPECIAL attribute. • Bit 2 is set by RACROUTE REQUEST=AUTH and RACROUTE REQUEST=DEFINE and indicates that the user has the OPERATIONS attribute and used this authority to obtain access to the resource. • Bit 3 indicates that the user has the AUDITOR or ROAUDIT attribute or group-AUDITOR and used this authority to issue the command with operands that require the AUDITOR or ROAUDIT attribute or group-AUDITOR authority. • Bit 4 indicates that the user has authority because the exit routine indicated that the request is to be accepted without any further authority checks. • Bit 5 indicates that resource access was granted by the operator during failsoft processing. • Bit 6 indicates that *BYPASS* was specified on the user ID field. Access was granted because RACF authority checking was bypassed. • Bit 7 indicates that the user has the trusted attribute. <p>2. SMF80REA: These flags indicate the reason RACF produced the SMF record.</p> <ul style="list-style-type: none"> • Bit 0 is set when there are changes made to a profile in a class specified in the AUDIT operand of the SETROPTS command. • Bit 1 is set when a user with the AUDITOR attribute specifies the UAUDIT operand on the ALTUSER command for a user and the user has changed RACF profiles with a RACF command, or a RACROUTE REQUEST=AUTH or RACROUTE REQUEST=DEFINE has been issued for the user. • Bit 2 is set when a user with the AUDITOR attribute specifies the SAUDIT or OPERAUDIT operand on the SETROPTS command and a user with either the SPECIAL or OPERATIONS attribute has changed RACF profiles with a RACF command. To determine whether SPECIAL or OPERATIONS authority was used, see the flags in SMF80ATH. Bit 1 indicates SPECIAL. Bit 2 indicates OPERATIONS. Note that if a user has both the AUDITOR attribute and either the SPECIAL or OPERATIONS attribute when issuing a command with operands that require only the AUDITOR attribute, RACF does not log this activity because the SPECIAL or OPERATIONS authority is not used. • Bit 3 is set for any of the following reasons: <ul style="list-style-type: none"> – The AUDIT option in the resource profile specifies that attempts to access the resource be logged – User audit bits for UNIX files and directories are set, which is done by using the chaudit command – The RACROUTE REQUEST=AUTH exit routine specifies unconditional logging – The console operator grants the resource access during failsoft processing. • Bit 4 is set when the RACROUTE REQUEST=VERIFY fails to verify a user because of an invalid group, password, terminal, or OIDCARD, or initACEE fails because a certificate is not defined or is not trusted. • Bit 5 is set if the RVARY or SETROPTS command produced the SMF record. (The execution of these two commands always produces an SMF record.) • Bit 6 is set when a user with the AUDITOR attribute specifies logging of command violations (with the CMDVIOL operand on the SETROPTS command) and RACF detects a violation. • Bit 7 is set when attempts to access a RACF-protected resource are being logged, as requested by the GLOBALAUDIT option in the resource profile. Or, for UNIX files and directories, if the AUDITOR audit bits are set (by using the chaudit -a command). <p>3. SMF80ERR: These flags indicate errors during command processing and the extent of the processing.</p> <ul style="list-style-type: none"> • Bit 0 indicates that an error occurred that prevented the command from completing all updates requested, and the command was unable to back out the updates already done. If this bit is on, there may be an inconsistency between the profiles on the RACF database, or between the profile for a data set and the RACF-indicator for the data set in the DSCB or catalog. The latter is also indicated by a bit in the command-related information for the ADDSD, ALTDSD, and DELDSD commands. For some commands (for example, ADDUSER), the inconsistency means an incompletely defined resource. For other commands, where the profiles are already defined (for example, ALTUSER), the inconsistency means that all changes were not made, but the profiles are still usable. This bit indicates a terminating error and should not be confused with a keyword violation or processing error where the command continues processing other operands. • Bit 1 indicates that none of the requested changes were made, because either (1) a terminating error occurred before the changes were made, or (2) the command was able to back out the changes after a terminating error. 					

Table of event codes and event code qualifiers

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

The event code qualifier is 0 if the recorded event is not a violation or a warning. There are exceptions for event code 1 (Job initiation/TSO logon/logoff); event qualifier codes 8, 12, 13 and 32 are not violations or warnings.

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

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

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

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

For detailed descriptions of the SMF event code qualifiers, see [Appendix E, “Event code qualifier descriptions,”](#) on page 655.

Event 1(1): JOB INITIATION / TSO LOGON/LOGOFF (detected by RACINIT request)		
Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP/ SMF80TP2 Values)
0(0)	Successful Initiation	1, 17, 20, 46, 47, 49, 53, 55, 331, 332, 374, 386, 392, 393, 394, 395, 424, 425, 443
1(1)	Password not valid	
2(2)	Group not valid	
3(3)	OIDCARD not valid	
4(4)	Terminal/console not valid	
5(5)	Application not valid	
6(6)	Revoked user attempting access	
7(7)	User ID automatically revoked because of excessive password and password phrase attempts.	
8(8)	Successful termination	
9(9)	Undefined user ID	
10(A)	Insufficient security label authority	
11(B)	Not authorized to security label	
12(C)	Successful RACINIT initiation	
13(D)	Successful RACINIT delete	
14(E)	System now requires more authority	
15(F)	Remote job entry - job not authorized	
16(10)	SURROGAT class is inactive	
17(11)	Submitter is not authorized by user	
18(12)	Submitter not authorized to security label	
19(13)	User is not authorized to job	
20(14)	WARNING - Insufficient security label authority	

Event 1(1): JOB INITIATION / TSO LOGON/LOGOFF (detected by RACINIT request)		
Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP/ SMF80TP2 Values)
21(15)	WARNING - security label missing from user, job, or profile	
22(16)	WARNING - not authorized to security label	
23(17)	Security labels not compatible	
24(18)	WARNING - security labels not compatible	
25(19)	Current PASSWORD has expired	
26(1A)	Invalid new PASSWORD	
27(1B)	Verification failed by installation	
28(1C)	Group access has been revoked	
29(1D)	OIDCARD is required	
30(1E)	Network job entry - job not authorized	
31(1F)	Warning - unknown user from trusted node propagated	
32(20)	Successful initiation using PassTicket	
33(21)	Attempted replay of PassTicket	
34(22)	Client security label not equivalent to server's	
35(23)	User automatically revoked because of inactivity	
36(24)	Password phrase is not valid	
37(25)	New password phrase is not valid	
38(26)	Current password phrase has expired	
39(27)	No RACF user ID found for distributed identity	
40(28)	Successful Multifactor Authentication (MFA)	
41(29)	Failed Multifactor Authentication (MFA)	
42(2A)	Failed authentication because no multifactor decision could be made for a MFA user who has the NOPWFALLBACK option.	
43(2B)	IBM MFA partial success: credentials were not incorrect, but a re-authentication is required.	
44(2C)	Identity Token validation error	
45(2D)	Identity Token build error	
46(2E)	Failed Identity Token authentication	

Event 2(2): RESOURCE ACCESS (detected by RACROUTE REQUEST=AUTH, RACROUTE REQUEST=FASTAUTH and DIRAUTH function)		
Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP/SMF80TP2 Values)
0(0)	Successful access	1, 3, 4, 5, 15, 16, 17, 20, 33, 38, 46, 48, 49, 50, 51, 53, 54, 55, 64, 65, 66, 331, 332, 386, 390 (see Notes 1 and 2), 392, 393, 394, 395, 396 (see Note 3), 424, 425, 445
1(1)	Insufficient authority	
2(2)	Profile not found - RACFIND specified on macro	
3(3)	Access permitted because of warning	
4(4)	Failed because of PROTECTALL	
5(5)	WARNING issued because of PROTECTALL	

Event 2(2): RESOURCE ACCESS (detected by RACROUTE REQUEST=AUTH, RACROUTE REQUEST=FASTAUTH and DIRAUTH function)		
Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP/SMF80TP2 Values)
6(6)	Insufficient CATEGORY/SECLEVEL	
7(7)	Insufficient security label authority	
8(8)	WARNING - security label missing from job, user, or profile	
9(9)	WARNING - insufficient security label authority	
10(A)	WARNING - Data set not cataloged	
11(B)	Data set not cataloged	
12(C)	Profile not found - required for authority checking	
13(D)	WARNING - insufficient CATEGORY/SECLEVEL	
14(E)	WARNING - Non-MAIN execution environment detected while in ENHANCED PGMSECURITY mode. Conditional access or use of EXECUTE-controlled program temporarily allowed.	
15(F)	Conditional access or use of EXECUTE-controlled program allowed through BASIC mode program while in ENHANCED PGMSECURITY mode.	
Notes: <ol style="list-style-type: none"> The SMF80DTP value 4 (access authority allowed) can be less than the SMF80DTP value 3 (access authority requested) in two cases: <ul style="list-style-type: none"> When RACF authorizes access to a user who requested access to a database because the user has the OPERATIONS attribute. When the RACROUTE REQUEST=AUTH exit routine returns a return code of 12, which indicates that the request should be granted. The SMF80DTP value of 16 appears only when the RACROUTE REQUEST=AUTH received an old volume (OLDVOL) as input. The value of 33 appears when a generic profile is used. Relocate 396 appears with event code qualifier 0. It appears only when access is granted because of the criteria entries on the conditional access list. 		

Event 3(3): ADDVOL/CHGVOL (detected by RACROUTE REQUEST=DEFINE TYPE=ADDVOL or CHGVOL)		
Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP/SMF80DA2 Values)
0(0)	Successful processing of new volume	1, 4, 5, 15, 16, 17, 33, 38, 44, 46, 49, 53, 51, 55, 331, 332, 386 (see Note), 392, 393, 394, 395, 424, 425
1(1)	Insufficient authority (DATASET only)	
2(2)	Insufficient security label authority	
3(3)	Less specific profile exists with different security label	
Note: The SMF80DTP value of 16 appears only when the RACROUTE REQUEST=AUTH received an old volume (OLDVOL) as input. The value of 33 appears when a generic profile is used.		

Event 4(4): RENAME RESOURCE (detected by RACROUTE REQUEST=DEFINE with TYPE=DEFINE and NEWNAME specified)		
Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP/SMF80DA2 Values)
0(0)	Successful rename	1, 2, 5, 15, 17, 33, 38, 44, 46, 49, 51, 53, 55, 331, 332, 386, 392, 393, 394, 395, 424, 425
1(1)	Group not valid	
2(2)	User not in group	
3(3)	Insufficient authority	

Event 4(4): RENAME RESOURCE (detected by RACROUTE REQUEST=DEFINE with TYPE=DEFINE and NEWNAME specified)		
Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP/ SMF80DA2 Values)
4(4)	Resource name already defined	
5(5)	User not defined to RACF	
6(6)	Resource not protected	
7(7)	WARNING - resource not protected	
8(8)	User in second qualifier is not RACF-defined	
9(9)	Less specific profile exists with different security label	
10(A)	Insufficient security label authority	
11(B)	Resource not protected by security label	
12(C)	New name not protected by security label	
13(D)	New security label must dominate old security label	
14(E)	Insufficient security label authority	
15(F)	WARNING - resource not protected by security label	
16(10)	WARNING - new name not protected by security label	
17(11)	WARNING - new security label must dominate old security label	
Note: In cases where the RACROUTE REQUEST=DEFINE is used to rename a resource (SMF80EVT=4), the data type 33 relocate section can hold a resource name that is either the old name or the new name, or it can hold the generic profile that protects the old or the new name.		

Event 5(5): DELETE RESOURCE (detected by RACROUTE REQUEST=DEFINE, TYPE=DELETE or DELETE)		
Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP/ SMF80DA2 Values)
0(0)	Successful scratch	1, 5, 15, 17, 33, 38, 44, 46, 49, 51, 53, 55, 331, 332, 386, 392, 393, 394, 395, 424, 425
1(1)	Resource not found	
2(2)	Invalid volume identification (DATASET only)	

Event 6(6): DELETE 1 VOLUME OF MULTIVOLUME RESOURCE (detected by RACROUTE REQUEST=DEFINE, TYPE=DELETE)		
Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP/ SMF80DA2 Values)
0(0)	Successful deletion	1, 5, 8, 15, 17, 38, 44, 46, 49, 51, 53, 55, 331, 332, 386, 392, 393, 394, 395, 424, 425

Event 7(7): DEFINE RESOURCE (detected by RACROUTE REQUEST=DEFINE, TYPE=DEFINE)		
Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP/ SMF80DA2 Values)
0(0)	Successful definition	1, 5, 15, 17, 18, 19, 33, 38, 44, 46, 49, 51, 53, 55, 331, 332, 386, 392, 393, 394, 395, 424, 425
1(1)	Group undefined	
2(2)	User not in group	
3(3)	Insufficient authority	

Event 7(7): DEFINE RESOURCE (detected by RACROUTE REQUEST=DEFINE, TYPE=DEFINE)		
Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP/ SMF80DA2 Values)
4(4)	Resource name already defined	
5(5)	User not defined to RACF	
6(6)	Resource not protected	
7(7)	WARNING - resource not protected	
8(8)	WARNING - security label missing from job, user, or profile	
9(9)	WARNING - insufficient security label authority	
10(A)	User in second qualifier is not RACF-defined	
11(B)	Insufficient security label authority	
12(C)	Less specific profile exists with a different security label	

EVENT dec(hex)	Command	Code qualifier dec(hex)	Description	Relocate type sections (possible SMF80DTP/ SMF80DA2 values)
8(8)	ADDSD	0(0)	No violations detected	6, 7, 10, 13, 33, 38, 40, 44, 49, 50, 51, 53, 55, 62, 63, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	
		3(3)	Successful retrieval of data set names affected by a security label change	
		4(4)	Error during retrieval of data set names affected by a security label change	
9(9)	ADDGROUP	0(0)	No violations detected	6, 7, 37, 38, 44, 49, 53, 55, 63, 301, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	
10(A)	ADDUSER	0(0)	No violations detected	6, 7, 8, 28, 37, 38, 40, 44, 49, 53, 55, 301, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	

EVENT dec(hex)	Command	Code qualifier dec(hex)	Description	Relocate type sections (possible SMF80DTP/ SMF80DA2 values)
11(B)	ALTDSD	0(0)	No violations detected	6, 7, 10, 11, 33, 38, 40, 41, 44, 49, 50, 51, 53, 55, 62, 63, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	
		3(3)	Successful retrieval of data set names affected by a security label change	
		4(4)	Error during retrieval of data set names affected by a security label change	
12(C)	ALTGROUP	0(0)	No violations detected	6, 7, 37, 38, 44, 49, 53, 55, 301, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	
13(D)	ALTUSER	0(0)	No violations detected	6, 7, 8, 28, 37, 38, 40, 41, 44, 49, 53, 55, 301, 331, 332, 386, 392, 393, 394, 395, 424, 425, 440, 441, 442
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	
14(E)	CONNECT	0(0)	No violations detected	6, 38, 49, 53, 55, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	
15(F)	DELDSD	0(0)	No violations detected	6, 38, 49, 50, 51, 53, 55, 62, 63, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	
		3(3)	Successful retrieval of data set names affected by a security label change	
		4(4)	Error during retrieval of data set names affected by a security label change	
16(10)	DELGROUP	0(0)	No violations detected	6, 38, 44, 49, 53, 55, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	

EVENT dec(hex)	Command	Code qualifier dec(hex)	Description	Relocate type sections (possible SMF80DTP/ SMF80DA2 values)
17(11)	DELUSER	0(0)	No violations detected	6, 38, 44, 49, 53, 55, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	
18(12)	PASSWORD	0(0)	No violations detected	6, 38, 49, 53, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	
19(13)	PERMIT	0(0)	No violation detected	6, 9, 12, 13, 14, 17, 26, 38, 39, 49, 53, 55, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Insufficient authority (partial or no update to RACF database; see SMF80ERR)	
20(14)	RALTER	0(0)	No violations detected	6, 7, 9, 10, 11, 17, 24, 25, 29, 33, 38, 40, 41, 44, 49, 50, 51, 53, 55, 301, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	
21(15)	RDEFINE	0(0)	No violations detected	6, 7, 9, 13, 17, 24, 29, 33, 38, 40, 44, 49, 50, 51, 53, 55, 301, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	
22(16)	RDELETE	0(0)	No violations detected	6, 9, 17, 38, 44, 49, 50, 51, 53, 55, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	
23(17)	REMOVE	0(0)	No violations detected	6, 17, 38, 49, 53, 55, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	
24(18)	SETROPTS	0(0)	No violations detected	6, 21, 22, 23, 27, 32, 34, 35, 36, 42, 43, 44, 45, 49, 53, 55, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	

EVENT dec(hex)	Command	Code qualifier dec(hex)	Description	Relocate type sections (possible SMF80DTP/ SMF80DA2 values)
25(19)	RVARY	0(0)	No violations detected	6, 27, 30, 31, 49, 53, 55, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial or no update to RACF database; see SMF80ERR)	
26(1A)	APPC SESSION ESTABLISHMENT	0(0)	Partner verification was successful	1, 17, 33, 38, 49, 53, 55, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Session established without verification	
		2(2)	Local LU key will expire in <= 5 days	
		3(3)	Partner LU access has been revoked	
		4(4)	Partner LU key does not match this LU key	
		5(5)	Session terminated for security reason	
		6(6)	Required SESSION KEY not defined	
		7(7)	Possible security attack by partner LU	
		8(8)	SESSION KEY not defined for partner LU	
		9(9)	SESSION KEY not defined for this LU	
		10(A)	SNA security-related protocol error	
		11(B)	Profile change during verification	
		12(C)	Expired SESSION KEY	
27(1B)	GENERAL	0(0)	General purpose auditing	17, 46, 49, 53, 55, 331, 332, 386, 392, 393, 394, 395, 424, 425
28(1C)	DIRECTORY SEARCH	0(0)	Access allowed	17, 49, 51, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 291, 295, 297, 298, 299, 307, 308, 309, 310, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Not authorized to search directory	
		2(2)	Security label failure	
29(1D)	CHECK ACCESS TO DIRECTORY	0(0)	Access allowed	17, 49, 51, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 297, 298, 299, 307, 308, 309, 310, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Caller does not have requested access authority	
		2(2)	Security label failure	
30(1E)	CHECK ACCESS TO FILE	0(0)	Access allowed	17, 49, 51, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 298, 299, 307, 308, 309, 310, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Caller does not have requested access authority	
		2(2)	Security label failure	

EVENT dec(hex)	Command	Code qualifier dec(hex)	Description	Relocate type sections (possible SMF80DTP/ SMF80DA2 values)
31(1F)	CHAUDIT	0(0)	File's audit options changed	17, 49, 51, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 292, 293, 294, 307, 308, 309, 310, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Caller does not have authority to change user audit options of specified file	
		2(2)	Caller does not have authority to change auditor audit options	
		3(3)	Security label failure	
32(20)	CHDIR	0(0)	Current working directory changed	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		*	Failures logged as directory search event types	
33(21)	CHMOD	0(0)	File's mode changed	17, 49, 51, 53, 55, 256, 257, 258, 259, 260, 261, 263, 264, 265, 266, 289, 290, 296, 307, 308, 309, 310, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Caller does not have authority to change mode of specified file	
		2(2)	Security label failure	
34(22)	CHOWN	0(0)	File's owner or group owner changed	17, 49, 51, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 280, 281, 307, 308, 309, 310, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Caller does not have authority to change owner or group owner of specified file	
		2(2)	Security label failure	
35(23)	CLEAR SETID BITS FOR FILE	0(0)	S_ISUID, S_ISGID, and S_ISVTX bits changed to zero (write)	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 289, 290, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
			No failure cases	
36(24)	EXEC WITH SETUID/SETGID	0(0)	Successful change of z/OS UNIX user identifiers (UIDs) and z/OS UNIX group identifiers (GIDs).	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 272, 273, 274, 275, 276, 277, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
			No failure cases. Access to program file is audited by an internal open	
37(25)	GETPSENT	0(0)	Access allowed	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 282, 283, 284, 288, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Not authorized to access specified process	
38(26)	INITIALIZE z/OS UNIX PROCESS (DUB)	0(0)	z/OS UNIX process successfully initiated	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	User not defined as a z/OS UNIX user (no user profile or no OMVS segment)	
		2(2)	User incompletely defined as a z/OS UNIX user (no z/OS UNIX user identifier (UID) in user profile)	
		3(3)	User's current group has no z/OS UNIX group identifier (GID).	
39(27)	z/OS UNIX PROCESS COMPLETION (UNDUB)	0(0)	Process completed	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
			No failure cases	

EVENT dec(hex)	Command	Code qualifier dec(hex)	Description	Relocate type sections (possible SMF80DTP/ SMF80DA2 values)
40(28)	KILL	0(0)	Access allowed	17, 49, 51, 53, 55, 256, 257, 258, 259, 260, 261, 262, 282, 283, 284, 288, 300, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Not authorized to access specified process	
		2(2)	Security label failure	
41(29)	LINK	0(0)	New link created	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 270, 299, 307, 308, 309, 310, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		*	Failures logged as directory search or check access event types	
42(2A)	MKDIR	0(0)	Directory successfully created	17, 49, 50, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 289, 290, 294, 296, 307, 308, 309, 310, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		*	Failures logged as directory search or check access event types	
43(2B)	MKNOD	0(0)	Node successfully created	17, 49, 50, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 289, 290, 294, 296, 307, 308, 309, 310, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		*	Failures logged as directory search or check access event types	
44(2C)	MOUNT FILE SYSTEM	0(0)	Successful mount	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 295, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		*	Failures logged as ck_priv event type	
45(2D)	OPEN (NEW FILE)	0(0)	File successfully created	17, 49, 50, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 289, 290, 294, 296, 307, 308, 309, 310, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		*	Failures logged as directory search or check access event types	
46(2E)	PTRACE	0(0)	Access allowed	17, 49, 51, 53, 55, 256, 257, 258, 259, 260, 261, 262, 282, 283, 284, 285, 286, 287, 288, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Not authorized to access specified process	
		2(2)	Security label failure	
47(2F)	RENAME	0(0)	Rename successful	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 270, 271, 278, 279, 294, 299, 302, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		*	Failures logged as directory search or check access event types	
48(30)	RMDIR	0(0)	Successful rmdir	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 307, 308, 309, 310, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		*	Failures logged as directory search or check access event types	
49(31)	SETEGID	0(0)	Successful change of effective z/OS UNIX group identifier (GID).	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 275, 276, 277, 281, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Not authorized to setegid	
50(32)	SETEUID	0(0)	Successful change of effective z/OS UNIX user identifier (UID).	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 272, 273, 274, 280, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Not authorized to seteuid	

EVENT dec(hex)	Command	Code qualifier dec(hex)	Description	Relocate type sections (possible SMF80DTP/ SMF80DA2 values)
51(33)	SETGID	0(0)	Successful change of z/OS UNIX group identifiers (GIDs).	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 275, 276, 277, 281, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Not authorized to setgid	
52(34)	SETUID	0(0)	Successful change of z/OS UNIX user identifiers (UIDs).	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 272, 273, 274, 280, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Not authorized to setuid	
53(35)	SYMLINK	0(0)	Successful symlink	17, 49, 50, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 297, 307, 308, 309, 310, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		*	Failures logged as directory search or check access event types	
54(36)	UNLINK	0(0)	Successful unlink	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 302, 307, 308, 309, 310, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		*	Failures logged as directory search or check access event types	
55(37)	UNMOUNT THE SYSTEM	0(0)	Successful unmount	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 295, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		*	Failures logged as ck_priv event type	
56(38)	CHECK FILE OWNER	0(0)	User is the owner	17, 49, 51, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 307, 308, 309, 310, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	User is not the owner	
		2(2)	Security label failure	
57(39)	CK_PRIV	0(0)	User is authorized	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	User is not authorized to use requested function	
58(3A)	OPEN SUBSIDIARY TTY	0(0)	Access allowed	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 282, 283, 284, 288, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Not authorized to access specified process	
59(3B)	RACLINK	0(0)	Access allowed	6, 49, 53, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority	
		2(2)	Keyword violation detected	
		3(3)	Association already defined	
		4(4)	Association already approved	
		5(5)	Association does not match	
		6(6)	Association does not exist	
		7(7)	Password not valid or user ID is revoked	
60(3C)	CHECK IPC ACCESS	0(0)	Access allowed	17, 49, 51, 56, 256, 257, 258, 259, 260, 261, 262, 265, 266, 267, 268, 269, 303, 304, 305, 306, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Caller does not have proper access authority	
		2(2)	Security label failure	

EVENT dec(hex)	Command	Code qualifier dec(hex)	Description	Relocate type sections (possible SMF80DTP/ SMF80DA2 values)
61(3D)	IPCGET (MAKE ISP)	0(0)	Successful creation of ISP	17, 49, 51, 56, 256, 257, 258, 259, 260, 261, 262, 265, 266, 269, 303, 304, 305, 306, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Security label failure	
62(3E)	R_IPC control	0(0)	Access allowed	17, 49, 51, 56, 256, 257, 258, 259, 260, 261, 262, 265, 266, 280, 281, 289, 290, 291, 296, 303, 304, 305, 306, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Caller does not have proper authority.	
		2(2)	Security label failure	
63(3F)	SETGROUP	0(0)	Access allowed	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Not authorized to access specified process	
64(40)	CHECK OWNER, TWO FILES	0(0)	User is the owner	17, 49, 51, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 271, 278, 279, 315, 316, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	User is not the owner	
		2(2)	Security label failure	
65(41)	R_AUDIT	0(0)	Successful r_audit	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
			No failure case	
66(42)	RACDCERT	0(0)	No violation detected	6, 49, 53, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 336, 337, 338, 339, 386, 392, 393, 394, 395, 398, 399, 424, 425, 446
		1(1)	Insufficient authority (no update to RACF database)	
67(43)	INITACEE	0(0)	Successful certificate registration	49, 53, 318, 319, 331, 332 374, 386, 392, 393, 394, 395, 424, 425, 446, 449
		1(1)	Successful certificate deregistration	
		2(2)	Not authorized to register the certificate	
		3(3)	Not authorized to unregister the certificate	
		4(4)	No user ID found for the certificate	
		5(5)	The certificate is not trusted	
		6(6)	Successful CERTAUTH certificate registration	
		7(7)	Insufficient authority to register the CERTAUTH certificate	
		8(8)	Client security label not equivalent to server's	
		9(9)	A SITE or CERTAUTH certificate was used to authenticate a user	
		10(A)	No RACF user ID found for distributed identity	
		11(B)	Successfully generated IDT from ACEE	
		12(C)	Failed attempting to generate IDT from ACEE	

EVENT dec(hex)	Command	Code qualifier dec(hex)	Description	Relocate type sections (possible SMF80DTP/ SMF80DA2 values)
68(44)	GRANT OF INITIAL KERBEROS TICKET (reserved for use by Network Authentication Service)	0(0)	Success	333, 334, 335
		1(1)	Failure	
69(45)	R_PKIServ GENCERT	0(0)	Successful GENCERT request	46, 49, 53, 318, 319, 331, 332, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 357, 358, 359, 373, 375, 376, 377, 378, 386, 388, 391, 392, 393, 394, 395, 422, 424, 425, 426, 427, 428
		1(1)	Insufficient authority for GENCERT	
		2(2)	Successful REQCERT request	
		3(3)	Insufficient authority for REQCERT	
		4(4)	Successful GENRENEW request	
		5(5)	Insufficient authority for GENRENEW	
		6(6)	Successful REQRENEW request	
		7(7)	Insufficient authority for REQNRENEW	
		8(8)	Successful PREREGISTER request	
		9(9)	Insufficient authority for PREREGISTER	
70(46)	R_PKIServ EXPORT	0(0)	Successful EXPORT request	46, 49, 53, 331, 332, 343, 344, 351, 359, 386, 391, 392, 393, 394, 395, 421, 424, 425
		1(1)	Insufficient authority for EXPORT	
		2(2)	Incorrect pass phrase specified for EXPORT	
71(47)	POLICY DIRECTOR ACCESS CONTROL DECISION (reserved for use by Policy Director Authorization Services)	0(0)	Authorized	352, 353, 354, 355, 356, 372
		1(1)	Not authorized but permitted because of warning mode	
		2(2)	Not authorized because of insufficient traverse authority but permitted because of warning mode	
		3(3)	Not authorized because of time- of-day check but permitted because of warning mode	
		4(4)	Not authorized	
		5(5)	Not authorized because of insufficient traverse authority	
		6(6)	Not authorized because of time- of-day check	
72(48)	R_PKIServ QUERY, DETAILS, or VERIFY	0(0)	Successful admin QUERY or DETAILS request	20, 46, 49, 53, 318, 319, 331, 332, 340, 341, 342, 346, 351, 358, 360, 361, 362, 363, 373, 375, 386, 391, 392, 393, 394, 395, 421, 422, 424, 425, 426, 429, 433, 434
		1(1)	Insufficient authority for admin QUERY or DETAILS	
		2(2)	Successful VERIFY request	
		3(3)	Insufficient authority for VERIFY	
		4(4)	Incorrect VERIFY certificate, no record found for this certificate	

EVENT dec(hex)	Command	Code qualifier dec(hex)	Description	Relocate type sections (possible SMF80DTP/ SMF80DA2 values)
73(49)	R_PKIServ UPDATEREQ	0(0)	Successful admin UPDATEREQ request	46, 49, 53, 331, 332, 340, 341, 342, 346, 347, 348, 349, 350, 351, 357, 364, 365, 375, 376, 377, 378, 386, 388, 391, 392, 393, 394, 395, 424, 425, 427, 428
		1(1)	Insufficient authority for admin UPDATEREQ	
74(4A)	R_PKIServ UPDATECERT or REVOKE	0(0)	Successful admin UPDATECERT request	48, 49, 53, 318, 331, 332,364, 365, 366, 386, 391, 392, 393, 394, 395, 423, 424, 425
		1(1)	Insufficient authority for admin UPDATECERT	
		2(2)	Successful REVOKE request	
		3(3)	Insufficient authority for REVOKE	
75(4B)	Change file ACL	0(0)	ACL successfully changed	17, 49, 51, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 307, 308, 309, 310, 315, 316, 317, 331, 332, 367, 368, 369, 370, 371, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority to change ACL	
		2(2)	Security label failure	
76(4C)	Remove file ACL	0(0)	Entire ACL removed	17, 49, 51, 53, 55, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 307, 308, 309, 310, 315, 316, 317, 331, 332, 367, 386, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority to remove ACL	
		2(2)	Security label failure	
77(4D)	Set file security label (R_setfsecl)	0(0)	Security label change successful	17, 49, 50, 51, 53, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Not authorized to change security label	
78(4E)	Set write-down privilege (R_writepriv)	0(0)	Requested function successful	49, 53, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Not authorized to IRR.WRITEDOWN.BYUSER	
79(4F)	CRL publication	0(0)	See z/OS Cryptographic Services PKI Services Guide and Reference .	
80(50)	RPKIRESP	0(0)	Successful RESPOND request	46, 49, 53, 331, 332, 386, 389, 391, 392, 393, 394, 395, 424, 425
		1(1)	Insufficient authority for RESPOND	
81(51)	PassTicket evaluation	0(0)	Success	20, 48, 49, 53, 67
		1(1)	Failure	
82(52)	PassTicket generation	0(0)	Success	20, 48, 49, 53, 67
		1(1)	Failure	

EVENT dec(hex)	Command	Code qualifier dec(hex)	Description	Relocate type sections (possible SMF80DTP/ SMF80DA2 values)
83(53)	RPKISCEP	0(0)	Successful AutoApprove PKCSReq request	46, 49, 53, 318, 319, 331, 332, 340, 341, 342, 346, 347, 348, 349, 350, 351, 357, 358, 359, 373, 375, 386, 388, 391, 392, 393, 394, 395, 424, 425, 427, 428
		1(1)	Successful AdminApprove PKCSReq request	
		2(2)	Successful GetCertInitial request	
		3(3)	Rejected PKCSReq or GetCertInitial request	
		4(4)	Incorrect SCEP transaction ID specified for GetCertInitial	
		5(5)	Insufficient authority for SCEPREQ	
84(54)	RDATAUPD	0(0)	Successful NewRing	49, 53, 318, 319, 320, 331, 332, 343, 344, 346, 386, 392, 393, 394, 395, 400, 401, 402, 403, 404, 405, 406, 407, 424, 425, 435, 436, 437, 438, 446
		1(1)	Not authorized to call NewRing	
		2(2)	Successful DataPut	
		3(3)	Not authorized to call DataPut	
		4(4)	Successful DataRemove	
		5(5)	Not authorized to call DataRemove	
		6(6)	Successful DelRing	
		7(7)	Not authorized to call DelRing	
85(55)	PKIAURNW	0(0)	Successful autoRenew	318, 319, 341, 342, 346, 358, 363, 373, 391, 408
86(56)	R_PgmSignVer	0(0)	Successful signature verification	1, 15, 46, 49, 53, 66, 331, 332, 386, 392, 393, 394, 395, 409, 410, 411, 412, 413, 414, 424, 425
		1(1)	Signature appears valid but root CA certificate not trusted	
		2(2)	Module signature failed verification	
		3(3)	Module certificate chain incorrect	
		4(4)	Signature required but module not signed	
		5(5)	Signature required but signature has been removed	
		6(6)	Program verification module not loaded. Program verification was not available when attempt was made to load this program.	
		7(7)	The algorithmic self-test failed while verifying the program verification module.	
87(57)	RACMAP	0(0)	No violation detected	6, 49, 53, 331, 332, 386, 392, 393, 394, 395, 415, 416, 424, 425
		1(1)	Insufficient authority (no update to RACF database)	
88(58)	AUTOPROF	0(0)	Successful profile modification	17, 49, 53, 55, 256, 257, 258, 259, 260, 261, 262, 317, 331, 332, 386, 392, 393, 394, 395, 417, 418, 419, 420, 424, 425

EVENT dec(hex)	Command	Code qualifier dec(hex)	Description	Relocate type sections (possible SMF80DTP/ SMF80DA2 values)
89(59)	RPKIQREC	0(0)	Successful user QRECOVER request	20, 46, 49, 53, 318, 319, 331, 332, 341, 342, 346, 358, 386, 391, 392, 393, 394, 395, 421, 424, 425
		1(1)	Insufficient authority for user QRECOVER	
90(5A)	PKIGENC	0(0)	Successful profile command	318, 319, 341, 342, 346, 391, 446, 447
91(5B)	PRLIMIT	0(0)	Access allowed	17, 49, 51, 53, 55, 256, 257, 258, 259, 260, 261, 262, 282, 283, 284, 285, 286, 287, 288, 317, 331, 332, 386, 392, 393, 394, 395, 424, 425
		1(1)	Not authorized to access specified process	
		2(2)	Security label failure	
92(5C)		0(0)	Security event	68

Table of relocate section variable data

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

Data type (SMF80DTP) dec(hex)	Data length (SMF80DLN)	Format	Description (SMF80DTA)
1(1)	1-255	EBCDIC	Resource name or old resource name (RACROUTE REQUEST=AUTH or RACROUTE REQUEST=DEFINE)
2(2)	1-255	EBCDIC	New data set name (RACROUTE REQUEST=DEFINE)
3(3)	1	Binary	Access requested (see Note “1” on page 65)
4(4)	1	Binary	Access allowed (see Note “2” on page 65)
5(5)	1	Binary	Data set level number (00-99)
6(6)	1-255	mixed	RACF command-related data (see “Table of data type 6 command-related data” on page 81)
7(7)	1-255	EBCDIC	DATA installation-defined data (ADDUSER, ALTUSER, RALTER, RDEFINE, ADDGROUP, ALTGROUP, ADDSD, ALTDSD)
8(8)	1-20	EBCDIC	NAME user-name (ADDUSER, ALTUSER)
9(9)	1-255	EBCDIC	Resource name (PERMIT, RALTER, RDEFINE, RDELETE)
10(A)	7	EBCDIC	Volume serial (ALTDSD ADDVOL, RALTER ADDVOL, ADDSD VOLUME). When set on, bit 0 of the first byte indicates that the volume was not processed. Bytes 2-7 contain the volume serial number.
11(B)	7	EBCDIC	Volume serial (ALTDSD DELVOL, RALTER DELVOL). When set on, bit 0 of the first byte indicates that the volume was not processed. Bytes 2-7 contain the volume serial.
12(C)	9-243		1 to 27 ID names (PERMIT), each 9 bytes long
		Binary	Byte 1: Processing flags: Bit Meaning when set 0 ID ignored because of processing error (see Note “3” on page 65) 1-7 Reserved for IBM's use
		EBCDIC	Bytes 2-9: ID name
13(D)	1-255	EBCDIC	FROM resource name (PERMIT, ADDSD, RDEFINE)
14(E)	12	EBCDIC	VOLUME volume serial (6 bytes) followed by FVOLUME volume serial (6 bytes) (PERMIT)

Data type (SMF80DTP) dec(hex)	Data length (SMF80DLN)	Format	Description (SMF80DTA)
15(F)	6	EBCDIC	VOLSER volume serial (RACROUTE REQUEST=AUTH or RACROUTE REQUEST=DEFINE) (Note that when RACROUTE REQUEST=AUTH receives a DATASET profile as input, the volume serial logged is the first volume serial contained in the profiles list of volume serials.)
16(10)	6	EBCDIC	OLDVOL volume serial (RACROUTE REQUEST=AUTH or RACROUTE REQUEST=DEFINE) (Note that when RACROUTE REQUEST=AUTH receives a DATASET profile as input, the volume serial logged is the first volume serial contained in the profiles list of volume serials.)
17(11)	1-8	EBCDIC	Class name (RACROUTE REQUEST=AUTH or RACROUTE REQUEST=DEFINE, RDEFINE, RALTER, RDELETE, PERMIT, or VMXEVENT auditing). For z/OS UNIX, class controlling auditing for the request.
18(12)	1-255	EBCDIC	MENTITY model resource name (RACROUTE REQUEST=DEFINE)
19(13)	6	EBCDIC	Volume serial of model resource (RACROUTE REQUEST=DEFINE)
20(14)	8	EBCDIC	Application name (RACROUTE REQUEST=VERIFY and VERIFYX)

Data type (SMF80DTP) dec(hex)	Data length (SMF80DLN)	Format	Description (SMF80DTA)
21(15)	10	binary	<p>Current class options (set by SETROPTS or RACF initialization)</p> <p>Byte 1:</p> <p>Bit</p> <p>Meaning when set</p> <p>0 Statistics are in effect</p> <p>1 Auditing is in effect</p> <p>2 Protection is in effect</p> <p>3 Generic profile processing is in effect</p> <p>4 Generic command processing is in effect</p> <p>5 Global access checking active</p> <p>6 RACLIST option in effect</p> <p>7 GENLIST option in effect</p>
		EBCDIC	<p>Bytes 2-9: Class name</p> <p>Byte 10:</p> <p>Bit</p> <p>Meaning when set</p> <p>0 Reserved for IBM's use</p> <p>1 LOGOPTIONS(ALWAYS) is in effect</p> <p>2 LOGOPTIONS(NEVER) is in effect</p> <p>3 LOGOPTIONS(SUCCESSSES) is in effect</p> <p>4 LOGOPTIONS(FAILURES) is in effect</p> <p>5 LOGOPTIONS(DEFAULT) is in effect</p> <p>6-7 Reserved for IBM's use</p>
22(16)	8	EBCDIC	Class name from STATISTICS/NOSTATISTICS keyword (SETROPTS)
23(17)	8	EBCDIC	Class name from AUDIT/NOAUDIT keyword (SETROPTS)
24(18)	2-247	EBCDIC	<p>Resource name from ADDMEM keyword (RDEFINE, RALTER)</p> <p>Byte 1:</p> <p>Bit</p> <p>Meaning when set</p> <p>0 Resource name not processed</p> <p>1 Resource name ignored because command user lacked sufficient authority to perform the operation</p> <p>Bytes 2-247: Resource name</p>

Data type (SMF80DTP) dec(hex)	Data length (SMF80DLN)	Format	Description (SMF80DTA)
25(19)	2-247	EBCDIC	Resource name from DELMEM keyword (RALTER). Bit 0 of the first byte, when set on, indicates that the resource name was not processed. Bytes 2-247 contain the resource name.
26(1A)	8	EBCDIC	Class name from FCLASS keyword (PERMIT)
27(1B)	8	EBCDIC	Class name from CLASSACT/NOCLASSACT keyword (SETROPTS, RVARY)
28(1C)	9	mixed	Class name from CLAUTH/NOCLAUTH keyword (ADDUSER, ALTUSER). Bit 1 of the first byte, when set on, indicates that the class was ignored because the command user did not have sufficient authority to perform the operation. Bytes 2-9 contain the class name.
29(1D)	1-255	EBCDIC	Application data (RDEFINE, RALTER)
30(1E)	12-55	mixed	<p>RACF database status (RVARY, RACF initialization)</p> <p>Byte 1:</p> <p>Bit</p> <p>Meaning when set</p> <p>0</p> <p>Database is active</p> <p>1</p> <p>Database is backup</p> <p>2-7</p> <p>Reserved for IBM's use</p> <p>Bytes 2-4: Unit name</p> <p>Bytes 5-10 Volume</p> <p>Byte 11: Sequence number</p> <p>Byte 12: 1-44 character data set name</p>
31(1F)	1-44	EBCDIC	Data set name from DATASET operand (RVARY)

Data type (SMF80DTP) dec(hex)	Data length (SMF80DLN)	Format	Description (SMF80DTA)
32(20)	89	mixed	<p>Byte</p> <p>Description</p> <p>1 Password interval value</p> <p>2 Password history value</p> <p>3 User ID revoke value</p> <p>4 Password warning level value</p> <p>5-84 Password syntax rules value</p> <p>85 User ID inactive interval</p> <p>86-89 Indicators</p> <p>Bit</p> <p>Meaning when set</p> <p>0 MODEL(GDG) in effect</p> <p>1 MODEL(USER) in effect</p> <p>2 MODEL(GROUP) in effect</p> <p>3 GRPLIST in effect</p> <p>4-31 Reserved for IBM's use</p>
33(21)	2-255	mixed	<p>Byte 1: Processing Flags</p> <p>Bit</p> <p>Meaning when set</p> <p>0 1=Resource name is generic</p> <p>0=Generic profile is used</p> <p>1 1=The old name of a data set renamed by RACROUTE REQUEST=DEFINE.</p> <p>0=The new name of a data set renamed by RACROUTE REQUEST=DEFINE.</p> <p>2-7 Reserved for IBM's use</p> <p>Bytes 2-254: Generic resource name or name of generic profile used</p> <p>Note: This relocate section does not appear in the record when a generic profile was not used, for example when a user is granted access to his own JES spool files without using a profile, even though one exists.</p>
34(22)	8	EBCDIC	Class name from GENERIC/NOGENERIC (SETROPTS)
35(23)	8	EBCDIC	Class name from GENCMD/NOGENCMD (SETROPTS)
36(24)	8	EBCDIC	Class name from GLOBAL/NOGLOBAL (SETROPTS)
37(25)	1-44	EBCDIC	Model name

Data type (SMF80DTP) dec(hex)	Data length (SMF80DLN)	Format	Description (SMF80DTA)
38(26)	8	EBCDIC	User ID or group name that owns the profile (RACROUTE REQUEST=AUTH and RACROUTE REQUEST=DEFINE and all the RACF commands that produce log records, except SETROPTS and RVARY). During DEFINE operations, this field contains the owner that the profile is defined with; in all other operations, it contains the current owner. Thus, for owner changes, it contains the old owner.
39(27)	4-255		Variable number of entity names (PERMIT), each 4 to 42 bytes long
		binary	Bytes 1-2: Processing flags: Bit Meaning when set 0 Entity ignored because of processing error 1 PROGRAM class entity 2 CONSOLE class entity 3 TERMINAL class entity 4 JESINPUT class entity 5 APPCPORT class entity 6 SYSID entity 7 SERVAUTH class entity 8 CRITERIA entity 9-15 Reserved for IBM's use Byte 3: Entity length
		EBCDIC	Bytes 4-end: Entity name
40(28)	2-45		Category name (ADDSD, ALTDSD, ADDUSER, ALTUSER, RDEFINE, RALTER commands and RACROUTE REQUEST=DEFINE) to be added to the profile, and organized as follows:
		binary	Byte 1 (at offset 0): Processing flags: Bit Meaning when set 0 Category name ignored because of processing error 1-7 Reserved for IBM's use
		EBCDIC	Bytes 2-end (at offset 1): Category name added
41(29)	2-45		Category name (ALTDSD, ALTUSER, and RALTER commands) to be deleted from the profile and organized as follows:
		binary	Byte 1 (at offset 0): Processing flags: Bit Meaning when set 0 Category name ignored because of processing error 1-7 Reserved for IBM's use
		EBCDIC	Bytes 2-end (at offset 1): Category name deleted

Data type (SMF80DTP) dec(hex)	Data length (SMF80DLN)	Format	Description (SMF80DTA)
42(2A)	8	EBCDIC	Class name from SETROPTS RACLIST/NORACLIST
43(2B)	1-8	EBCDIC	Class name from SETROPTS GENLIST/NOGENLIST
44(2C)	1-255	mixed	<p>Any segment data, except BASE</p> <p>Byte 1:</p> <p>Bit</p> <p>Meaning when set</p> <p>0 Reserved for IBM's use</p> <p>1 Delete the segment</p> <p>2-7 Reserved for IBM's use</p> <p>Byte 2-9: Name of segment</p> <p>Byte 10: Length of subkeyword</p> <p>Variable length The subkeyword specified</p> <p>Variable length The value associated with the subkeyword (limited to 245 minus length of subkeyword)</p>
44(2C)	1-255	mixed	<p>Directed command information</p> <p>Byte</p> <p>Description</p> <p>1 Bit string</p> <p>2-9 Name of segment - CMDSRC</p> <p>10 Length of subkeyword - 15</p> <p>11-25 Subkeyword ORIGINATED_FROM</p> <p>Variable length Contains one of the following:</p> <ul style="list-style-type: none"> • node.userid.DIRECTED_BY_AT • node.userid.DIRECTED_BY_ONLYAT • node.userid.DIRECTED_AUTOMATICALLY
44(2C)	1-255	mixed	<p>Directed application update information</p> <p>Byte</p> <p>Description</p> <p>1 Bit string</p> <p>2-9 Name of segment - APPLSRC</p> <p>10 Length of subkeyword - 15</p> <p>11-25 Subkeyword ORIGINATED_FROM</p> <p>Variable length node.userid.DIRECTED_AUTOMATICALLY</p>

Data type (SMF80DTP) dec(hex)	Data length (SMF80DLN)	Format	Description (SMF80DTA)
45(2D)	9		Class and logging options from SETROPTS LOGOPTIONS
		EBCDIC	Bytes 1-8: Class name
		mixed	Byte 9: Bit Meaning when set 0 ALWAYS 1 NEVER 2 SUCCESSES 3 FAILURES 4 DEFAULTS 5-7 Reserved for IBM's use
46(2E)	1-255	EBCDIC	Variable length string of data specified on LOGSTR= keyword on RACROUTE macro. Note: The log string specified on RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX is propagated to the port of entry authorization check made in the SERVAUTH class performed by VERIFY/X when SERVAUTH= is specified by the caller.
47(2F)	8	EBCDIC	JOBNAME that user is not authorized to submit for a JESJOBS job
48(30)	8	EBCDIC	User ID to whom data is directed (RECV= keyword on RACROUTE macro)
49(31)	1-20	EBCDIC	User name from ACEE
50(32)	8	EBCDIC	Security label name (ADDSD, ALTDSD, ALTUSER, RDEFINE, and RALTER commands, and the R_setfsecl, makeFSP and makeISP callable services) to be added to the profile or security packet, or the user security label for RACROUTE REQUEST=DIRAUTH
51(33)	8	EBCDIC	Security label name (RACROUTE REQUEST=AUTH and DIRAUTH, ck_access, ck_IPC_access, R_IPC_ctl, R_chmod, R_chown, R_audit, R_setfacl, ck_file_owner, ck_owner_two_files, ck_process_owner, R_ptrace or VMXEVENT auditing) of the resource, or security label name (ALTDSD, ALTUSER, RALTER commands and the R_setfsecl callable service) to be deleted from the profile or security packet.
53(35)	80	mixed	User security token, see "RUTKN" in <i>z/OS Security Server RACF Data Areas</i> in the <i>z/OS Internet library</i> (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary).
54(36)	80	mixed	Resource security token (RACROUTE REQUEST=AUTH) see "RUTKN" in <i>z/OS Security Server RACF Data Areas</i> in the <i>z/OS Internet library</i> (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary).
55(37)	8	Binary	Key to link audit records together
62(3E)	1-44	EBCDIC	Data set name affected by a security label change (used by SMF type 83 records)
63(3F)	4	EBCDIC	Link value to connect data sets affected by a security label change with the RACF command that caused the change
64(40)	4	EBCDIC	Link value to connect client and server audit records. A link value can appear for a client or server without a corresponding link value if: <ul style="list-style-type: none"> • The client has failed authorization • Auditing is not performed for both users

Data type (SMF80DTP) dec(hex)	Data length (SMF80DLN)	Format	Description (SMF80DTA)
65(41)	1	Binary	Flags that indicate ACEE type: Bit Meaning when set 0-4 Reserved for IBM's use 5 1=Nested ACEE 6 0=Reserved for IBM's use 1=Server 7 0=Unauthenticated client 1=Authenticated client
66(42)	44	EBCDIC	Partitioned data set name
67(43)	variable	mixed	Byte 1: PassTicket Generation or Evaluation Details Bit Meaning when set 0 Legacy PassTicket 1 Evaluation: Legacy PassTicket Successful 2 Enhanced PassTicket Type UPPER 3 Evaluation: Enhanced PassTicket Type UPPER Successful 4 Enhanced PassTicket Type MIXED 5 Evaluation: Enhanced PassTicket Type MIXED Successful 6 Evaluation: Failure due to PassTicket replay attempt 7 Reserved Byte 2: Reserved Byte 3-6: Return Code Byte 7-10: Reason Code Byte 11-18: Application Name
68(44)	16	EBCDIC	User and module

Data type (SMF80DTP) dec(hex)	Data length (SMF80DLN)	Format	Description (SMF80DTA)
Notes: 1. The access flags are: Bit - Access authority 0 - ALTER 1 - CONTROL 2 - UPDATE 3 - READ 4 - NONE 5 - Reserved for IBM's use 6 - WRITE (for REQUEST=DIRAUTH only). For RACROUTE REQUEST=DIRAUTH, bits 3 and 6 can both be on, indicating READWRITE authority. 2. The access flags for RACROUTE REQUEST=DIRAUTH are: Bit Access type 0 - Always on 1 - Mandatory access check 2 - Reverse mandatory access check 3 - Equal mandatory access check The access flags for other RACROUTE REQUEST types are: Bit Access authority 0 - ALTER 1 - CONTROL 2 - UPDATE 3 - READ 4 - NONE 5 - EXECUTE The access flags could all be off if a mandatory access check has failed. 3. This bit is turned on for each ID in the list (data type 12) and each program entity name in the list (data type 39) that was not processed because of a non-terminating error, such as user IDs (specified on the ID operand of the PERMIT command) that are not defined to RACF. If a terminating error, such as a RACF manager error, occurred while processing an ID or entity, this bit is turned on for all remaining IDs or entities that were not processed. For the PERMIT DELETE command, when no terminating error has occurred, this bit is turned ON only if no entry in the access list was deleted for the ID or entity. The access flags for other RACROUTE REQUEST types are: Bit Access authority 0 - ALTER 1 - CONTROL 2 - UPDATE 3 - READ 4 - NONE 5 - EXECUTE The access flags could all be off if a mandatory access check has failed.			

Table of extended-length relocate section variable data

This table describes the variable data elements of the extended-length relocate section.

Table 3. Table of extended-length relocate section variable data				
Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
256(100)	2	Binary	All	Audit function code, indicating the calling service. Refer to the description of IRRPAFC in z/OS RACF Data Areas in the z/OS Internet library (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary).

Table 3. Table of extended-length relocate section variable data (continued)

Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
257(101)	4	Binary	All	Old real z/OS UNIX user identifier (UID)
258(102)	4	Binary	All	Old effective z/OS UNIX user identifier (UID)
259(103)	4	Binary	All	Old saved z/OS UNIX user identifier (UID)
260(104)	4	Binary	All	Old real z/OS UNIX group identifier (GID)
261(105)	4	Binary	All	Old effective z/OS UNIX group identifier (GID)
262(106)	4	Binary	All	Old saved z/OS UNIX group identifier (GID)
263(107)	1-1023	EBCDIC	28,29,30,31,32, 33,34,35,41,42, 43,44,45,47,48, 53,54,55,56,64	Requested path name (see also data type 299) Note: For events 47 (rename) and 41 (link), this is the old path name.
264(108)	16	Binary	28,29,30,31,32, 33,34,35,41,42, 43,44,45,47,48, 53,54,55,56,64	File identifier
265(109)	4	Binary	28,29,30,31,32, 33,34,35,41,42, 43,44,45,47,48, 53,54,55,56,64	File owner z/OS UNIX user identifier (UID)
265(109)	4	Binary	60,61,62	IPC key owner z/OS UNIX user identifier (UID)
266(10A)	4	Binary	28,29,30,31,32, 33,34,35,41,42, 43,44,45,47,48, 53,54,55,56,64	File owner z/OS UNIX group identifier (GID)
266(10A)	4	Binary	60,61,62	IPC key owner z/OS UNIX group identifier (GID)
267(10B)	1	Binary	28,29,30	Requested access Value Meaning X'04' Read access X'02' Write access X'01' Execute access X'81' Directory search access X'87' Any access Multiple bits may be set.
267(10B)	1	Binary	60	IPC requested access Value Meaning X'00' No access X'02' Write access X'04' Read access X'06' Read and write access

Table 3. Table of extended-length relocate section variable data (continued)

Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
268(10C)	1	Binary	28, 29, 30, 60	<p>Access type (bits used to make access check)</p> <p>Value Meaning</p> <p>1 'owner' bits</p> <p>2 'group' bits</p> <p>3 'other' bits</p> <p>4 no bits used</p> <p>5 UID ACL entry</p> <p>6 GID ACL entry or entries</p> <p>7 ACL exists but could not be retrieved</p> <p>8 A restricted user ID was denied access because it was not the file owner and was not explicitly permitted to the file</p> <p>The access type value could be 0 if a mandatory access check has failed.</p>
269(10D)	1	Binary	28,29,30	<p>Access allowed</p> <p>Value Meaning</p> <p>X'04' Read access</p> <p>X'02' Write access</p> <p>X'01' execute/search</p> <p>Multiple bits can be set.</p>
269(10D)	1	Binary	60	<p>IPC access allowed</p> <p>Value Meaning</p> <p>X'02' Write access</p> <p>X'04' Read access</p> <p>Multiple bits can be set.</p>
270(10E)	1-1023	EBCDIC	28,29,30,41,47	<p>Second requested path name (see also data type 299)</p> <p>Note: For events 47 (rename) and 41 (link), this is the new path name.</p>
271(10F)	16	Binary	47,64	Second file identifier
272(110)	4	Binary	36,50,52	New real z/OS UNIX user identifier (UID)
273(111)	4	Binary	36,50,52	New effective z/OS UNIX user identifier (UID)
274(112)	4	Binary	36,50,52	New saved z/OS UNIX user identifier (UID)
275(113)	4	Binary	36,49,51	New real z/OS UNIX group identifier (GID)
276(114)	4	Binary	36,49,51	New effective z/OS UNIX group identifier (GID)
277(115)	4	Binary	36,49,51	New saved z/OS UNIX group identifier (GID)
278(116)	4	Binary	47	Owner z/OS UNIX user identifier (UID) of deleted file
278(116)	4	Binary	64	Second file owner z/OS UNIX user identifier (UID)
279(117)	4	Binary	47	Owner z/OS UNIX group identifier (GID) of deleted file

Table 3. Table of extended-length relocate section variable data (continued)				
Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
279(117)	4	Binary	64	Second file owner z/OS UNIX group identifier (GID)
280(118)	4	Binary	34,50,52	z/OS UNIX user identifier (UID) input parameter
280(118)	4	Binary	62	IPC owner z/OS UNIX user identifier (UID) input parameter
281(119)	4	Binary	34,49,51	z/OS UNIX group identifier (GID) input parameter
281(119)	4	Binary	62	IPC owner z/OS UNIX group identifier (GID) input parameter
282(11A)	4	Binary	37,40,46,58	Target real z/OS UNIX user identifier (UID)
283(11B)	4	Binary	37,40,46,58	Target effective z/OS UNIX user identifier (UID)
284(11C)	4	Binary	37,40,46,58	Target saved z/OS UNIX user identifier (UID)
285(11D)	4	Binary	46	Target real z/OS UNIX group identifier (GID)
286(11E)	4	Binary	46	Target effective z/OS UNIX group identifier (GID)
287(11F)	4	Binary	46	Target saved z/OS UNIX group identifier (GID)
288(120)	4	Binary	37,40,46,58	Target PID
289(121)	4	Binary	33,35	<p>Old mode</p> <p>Bit Meaning</p> <p>0-19 Reserved for IBM's use</p> <p>20 S_ISGID bit</p> <p>21 S_ISUID bit</p> <p>22 S_ISVTX bit</p> <p>23-25 Owner permission bits (read/write/execute)</p> <p>26-28 Group permission bits (read/write/execute)</p> <p>29-31 Other permission bits (read/write/execute)</p>
289(121)	4	Binary	62	<p>IPC old mode</p> <p>Bit Meaning</p> <p>0-22 Reserved for IBM's use</p> <p>23-25 Owner permission bits (RW-)</p> <p>26-28 Group permission bits (RW-)</p> <p>29-31 Other permission bits (RW-)</p>

Table 3. Table of extended-length relocate section variable data (continued)

Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
290(122)	4	Binary	33,35,42,43,45	<p>New mode</p> <p>Bit</p> <p>Meaning</p> <p>0-19 Reserved for IBM's use</p> <p>20 S_ISGID bit</p> <p>21 S_ISUID bit</p> <p>22 S_ISVTX bit</p> <p>23-25 Owner permission bits (read/write/execute)</p> <p>26-28 Group permission bits (read/write/execute)</p> <p>29-31 Other permission bits (read/write/execute)</p>
290(122)	4	Binary	62	<p>IPC new mode</p> <p>Bit</p> <p>Meaning</p> <p>0-22 Reserved for IBM's use</p> <p>23-25 Owner permission bits (RW-)</p> <p>26-28 Group permission bits (RW-)</p> <p>29-31 Other permission bits (RW-)</p>
291(123)	2	Binary	28	Service that was being processed. Used when data type 256 indicates that the calling service was lookup (path name resolution).
291(123)	2	Binary	62	Service that was being processed. Used when data type 256 indicates that the calling service was to remove an ID, set, or setmqb.
292(124)	4	Binary	31	<p>Requested audit options</p> <p>Byte</p> <p>Meaning</p> <p>1 Read access audit options</p> <p>2 Write access audit options</p> <p>3 execute/search audit options</p> <p>4 Reserved for IBM's use</p> <p>In each byte, the following flags are defined:</p> <p>Value</p> <p>Meaning</p> <p>X'00' Do not audit any access attempts</p> <p>X'01' Audit successful accesses</p> <p>X'02' Audit failed access attempts</p> <p>X'03' Audit both successful and failed access attempts</p>

Table 3. Table of extended-length relocate section variable data (continued)

Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
293(125)	8	Binary	31	<p>Old audit options (user and auditor)</p> <p>Byte</p> <p>Meaning</p> <p>1 User read access audit options</p> <p>2 User write access audit options</p> <p>3 User execute/search audit options</p> <p>4 Reserved for IBM's use</p> <p>5 Auditor read access audit options</p> <p>6 Auditor write access audit options</p> <p>7 Auditor execute/search audit options</p> <p>8 Reserved for IBM's use</p> <p>In each byte, the following flags are defined:</p> <p>Value</p> <p>Meaning</p> <p>X'00' Do not audit any access attempts</p> <p>X'01' Audit successful accesses</p> <p>X'02' Audit failed access attempts</p> <p>X'03' Audit both successful and failed access attempts</p>
294(126)	8	Binary	31	<p>New audit options (user and auditor)</p> <p>Byte</p> <p>Meaning</p> <p>1 User read access audit options</p> <p>2 User write access audit options</p> <p>3 User execute/search audit options</p> <p>4 Reserved for IBM's use</p> <p>5 Auditor read access audit options</p> <p>6 Auditor write access audit options</p> <p>7 Auditor execute/search audit options</p> <p>8 Reserved for IBM's use</p> <p>In each byte, the following flags are defined:</p> <p>Value</p> <p>Meaning</p> <p>X'00' Do not audit any access attempts</p> <p>X'01' Audit successful accesses</p> <p>X'02' Audit failed access attempts</p> <p>X'03' Audit both successful and failed access attempts</p>

Table 3. Table of extended-length relocate section variable data (continued)

Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
295(127)	1-44	EBCDIC	28,44,55	Data set name for mounted file system
296(128)	4	Binary	33,42,43,45	<p>Requested file mode</p> <p>Bit</p> <p>Meaning</p> <p>0-19 Reserved for IBM's use</p> <p>20 S_ISGID bit</p> <p>21 S_ISUID bit</p> <p>22 S_ISVTX bit</p> <p>23-25 Owner permission bits (read/write/execute)</p> <p>26-28 Group permission bits (read/write/execute)</p> <p>29-31 Other permission bits (read/write/execute)</p>
296(128)	4	Binary	61,62	<p>IPC requested ISP mode.</p> <p>Bit</p> <p>Meaning</p> <p>0-22 Reserved for IBM's use</p> <p>23-25 Owner permission bits (RW-)</p> <p>26-28 Group permission bits (RW-)</p> <p>29-31 Other permission bits (RW-)</p>
297(129)	1-1023	EBCDIC	28,29,53	Content of symlink
298(12A)	1-256	EBCDIC	28,29,30	File name being checked
299(12B)	1	Binary	28,29,30, 41,47	<p>Flag indicating whether the requested path name is the old (or only) path name or the new path name. This field is X'01' except for ck_access events where authority to a new name is being checked. The second path name contains the new name specified.</p> <p>Value</p> <p>Meaning</p> <p>X'01' Old (or only) path name</p> <p>X'02' New path name</p>
300(12C)	4	Binary	40	Kill signal code

Table 3. Table of extended-length relocate section variable data (continued)

Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
301(12D)	variable	EBCDIC	9,10,12,13	<p>Command segment data</p> <p>Bytes 1-2</p> <p>Bit</p> <p>Meaning when set</p> <p>0</p> <p>Keyword was ignored because of insufficient authority</p> <p>1</p> <p>Segment is to be deleted by using a NOxxx keyword</p> <p>2-3</p> <p>Data format</p> <p>01</p> <p>Numeric</p> <p>10</p> <p>Hex</p> <p>11</p> <p>Undefined</p> <p>4</p> <p>Keyword has no subfield</p> <p>5-15</p> <p>Reserved for IBM's use</p> <p>Bytes 3-10: Name of segment (main keyword)</p> <p>Byte 11: Length of subkeyword; 0 if byte 1 bit 1 is set</p> <p>Variable length: The subkeyword specified; null if byte 1 bit 1 is set</p> <p>2 bytes: Length of data</p> <p>Variable length: The data as entered on the command</p>
302(12E)	1	Binary	47,54	<p>Last link deleted flag</p> <p>Value</p> <p>Meaning</p> <p>X'00'</p> <p>Last link was not deleted</p> <p>X'01'</p> <p>Last link was deleted.</p>
303(12F)	4	Binary	60,61,62	IPC key
304(130)	4	Binary	60,61,62	IPC ID
305(131)	4	Binary	60,61,62	IPC key creator z/OS UNIX user identifier (UID)
306(132)	4	Binary	60,61,62	IPC key creator z/OS UNIX group identifier (GID)
307(133)	8	EBCDIC	28,29,30,31,33, 34,41,42,43,45, 47,48,53,54,56	Filepool name
308(134)	8	EBCDIC	28,29,30,31,33, 34,41,42,43,45, 47,48,53,54,56	Filespace name
309(135)	4	Binary	28,29,30,31,33, 34,41,42,43,45, 47,48,53,54,56	Inode (file serial number)
310(136)	4	Binary	28,29,30,31,33, 34,41,42,43,45, 47,48,53,54,56	SCID (file serial number)
311(137)	8	EBCDIC	47	Second filepool name
312(138)	8	EBCDIC	47	Second filesystem name
313(139)	4	Binary	47	Second Inode (file serial number)
314(13A)	4	Binary	47	Second SCID (file serial number)

Table 3. Table of extended-length relocate section variable data (continued)

Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
315(13B)	4	EBCDIC	28,29,30,31,32, 33,34,41,44,47, 48,54,55,56,57, 63,64	Link value to connect client and server audit records. A link value may appear for a client or server without a corresponding link value if: <ul style="list-style-type: none"> • Client has failed authorization • Auditing is not performed for both users
316(13C)	1	Binary	28,29,30,31,32, 33,34,41,44,47,48,54, 55,56,57,63,64	Flags that indicate ACEE type: <p>Bit</p> <p>Meaning when set</p> <p>0–4 Reserved for IBM's use</p> <p>5 1=Nested ACEE</p> <p>6 0=Reserved for IBM's use</p> <p>1=Server</p> <p>7 0=Unauthenticated client</p> <p>1=Authenticated client</p>
317(13D)	1	Binary	28,29,30,31,32, 33,34,35,36,37, 38,39,40,41,42, 43,44,45,46,47, 48,49,50,51,52, 53,54,55,56,57, 58,60,61,62,63, 64,65	Value Meaning X'80' Indicates a default z/OS UNIX security environment is in effect.
318(13E)	1-255	EBCDIC	66, 67, 69, 72, 74, 79, 83, 85, 89	Certificate or CRL serial number
319(13F)	1-255	EBCDIC	66, 67, 69, 72, 74, 79, 83, 85, 89	Certificate or CRL issuer's distinguished name
320(140)	1-237	Char	66	Ring name
321(141)	1-64	Char	66	C from SUBJECTSDN
322(142)	1-64	Char	66	SP from SUBJECTSDN
323(143)	1-64	Char	66	L from SUBJECTSDN
324(144)	1-64	Char	66	O from SUBJECTSDN
325(145)	1-64	Char	66	OU from SUBJECTSDN
326(146)	1-64	Char	66	T from SUBJECTSDN
327(147)	1-64	Char	66	CN from SUBJECTSDN
328(148)	1-255	EBCDIC	66	SDNFILTER filter name
329(149)	1-255	EBCDIC	66	IDNFILTER filter name
330(14A)	1-255	EBCDIC	66	CRITERIA or NEWCRITERIA value
331(14B)	1-255	EBCDIC	ALL events except 68	Subject's distinguished name
332(14C)	1-255	EBCDIC	ALL events except 68	Issuer's distinguished name
333(14D)	1-240	EBCDIC	68	Kerberos principal name (reserved for use by Network Authentication Service)
334(14E)	7-22	EBCDIC	68	Kerberos login request source (reserved for use by Network Authentication Service)
335(14F)	1-10	EBCDIC	68	Kerberos KDC status code (reserved for use by Network Authentication Service)
336(150)	1-255	EBCDIC	66	ALTNAME IP address. This field might be repeated.
337(151)	1-255	EBCDIC	66	ALTNAME email address. This field might be repeated.
338(152)	1-255	EBCDIC	66	ALTNAME domain name. This field might be repeated.
339(153)	1-255	EBCDIC	66	ALTNAME URI. This field might be repeated.

Table 3. Table of extended-length relocate section variable data (continued)

Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
340(154)	1	Binary	69, 83	IRRSPX00 flags byte 1 – KeyUsage flag combinations: Bits Meaning 1... "handshake" (digitalsig, keyencrypt) .1.. "dataencrypt" ..1. "certsign" (keycertsign, crlsign) ...1 "docsign" 1... "keyagree"1.. "digitalsig"1. "keycertsign" 1... .1.. "keyencrypt" ..1. ..1. "crlsign"
341(155)	10	EBCDIC	69, 83, 85, 89	Requested NotBefore field in the format yyyy/mm/dd
342(156)	10	EBCDIC	69, 83, 85, 89	Requested NotAfter field in the format yyyy/mm/dd
343(157)	8	EBCDIC	69, 70	IRRSPX00 target user ID
344(158)	1-32	EBCDIC	69, 70	IRRSPX00 target label
345(159)	1-45	EBCDIC	69	IRRSPX00 SignWith field
346(15A)	1-255	EBCDIC	69, 83, 85, 89	Requested Subject's DN
347(15B)	1-64	EBCDIC	69, 83	Requested AltIPAddr field
348(15C)	1-255	EBCDIC	69, 83	Requested AltURI field
349(15D)	1-100	EBCDIC	69, 83	Requested AltEmail field
350(15E)	1-100	EBCDIC	69, 83	Requested AltDomain field
351(15F)	1-56	EBCDIC	69, 70, 83	IRRSPX00 CertId
352(160)	1-4096	EBCDIC	71	Policy Director protected object (reserved for use by Policy Director Authorization Services)
353(161)	1-1024	EBCDIC	71	Requested Policy Director permissions (reserved for use by Policy Director Authorization Services)
354(162)	8	EBCDIC	71	Policy Director principal user ID (reserved for use by Policy Director Authorization Services)
355(163)	36	EBCDIC	71	Principal ID string in the format <i>nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnn</i> where <i>n</i> is any hexadecimal digit (reserved for use by Policy Director Authorization Services)
356(164)	4	Binary	71	Policy Director quality of protection value (reserved for use by Policy Director Authorization Services)
357(165)	1024	EBCDIC	69, 70, 73, 83	HostIDMappings extension data
358(166)	1-32	EBCDIC	70, 83, 85, 89	Certificate requester's name
359(167)	1	Binary	69, 70, 83	IRRSPX00 flags byte 2 Bit Meaning 0 Pass phrase specified

Table 3. Table of extended-length relocate section variable data (continued)

Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
360(168)	32	EBCDIC	72	Certificate or certificate request status: <ul style="list-style-type: none"> • Pending approval • Approved • Completed • Rejected • Rejected, User Notified • Active • Expired • Revoked • Revoked, Expired
361(169)	10	EBCDIC	72	Creation date in the format yyyy/mm/dd
362(16A)	10	EBCDIC	72	Last modified in the format yyyy/mm/dd
363(16B)	1–255	EBCDIC	72, 85	Certificate serial number for previously issued certificate
364(16C)	4	Binary	73, 74	Action that is taken on certificate or certificate request
365(16D)	1–64	EBCDIC	74	Action comment
366(16E)	4	Binary	74	Certificate revocation reason
367(16F)	1	Binary	75, 76	ACL type <p>Value</p> <p>Meaning</p> <p>X'80' Access ACL</p> <p>X'40' File model</p> <p>X'20' Directory model</p>
368(170)	1	Unsigned	75	Effective ACL entry operation type <p>Value</p> <p>Meaning</p> <p>1 Add</p> <p>2 Modify</p> <p>3 Delete</p>
369(171)	5	Binary	75	ACL entry identifier. This consists of a 1–byte type code followed by the 4–byte hexadecimal UID or GID value. <p>Value</p> <p>Meaning</p> <p>X'01' User (UID) entry</p> <p>X'02' Group (GID) entry</p>
370(172)	1	Binary	75	Old ACL entry bits for modify and delete operations.
371(173)	1	Binary	75	New ACL entry bits for add and modify operations.
372(174)	1	Binary	71	Policy Director credential type flag reserved for use by Policy Director Authorization Services <p>Value</p> <p>Meaning</p> <p>X'00' Unauthenticated</p> <p>X'01' Authenticated</p>

Table 3. Table of extended-length relocate section variable data (continued)				
Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
373(175)	1-64	EBCDIC	69, 72, 83, 85	Email address for notification purposes
374(176)	8	EBCDIC	1, 67	Server's security label
375(177)	1-255	EBCDIC	69, 72, 73, 83	Extended keyUsage
376(178)	1-32	EBCDIC	69, 73	Certificate policies
377(179)	1-1024	EBCDIC	69, 73	Authority information access
378(17A)	1-255	EBCDIC	69, 73	Critical extensions
379(17B)	1-255	EBCDIC	79	CRL's issuing distribution point DN
380(17C)	10	EBCDIC	79	CRL's date of issue
381(17D)	8	EBCDIC	79	CRL's time of issue
382(17E)	10	EBCDIC	79	CRL's expiration date
383(17F)	8	EBCDIC	79	CRL's expiration time
384(180)	10	EBCDIC	79	CRL's date of publish
385(181)	8	EBCDIC	79	CRL's time of publish
386(182)	1-64	EBCDIC	All, except 68, 71, 79, and 85	SERVAUTH port of entry name (profile name protecting the SERVAUTH name if resource name is unavailable)
387(183)	1-1024	EBCDIC	79	CRL's issuing distribution point URI
388(184)	1-1024	EBCDIC	69, 73, 83	Requested ALTNAME OtherName
389(185)	1-1024	EBCDIC	80	Response from OSCP responder containing a list of triplets: <ul style="list-style-type: none"> • Certificate serial number • Status: GOOD, REVOKED, or UNKNOWN • Issuer's DN, or "UNKNOWN ISSUER" Each item is separated by a comma and each triplet is separated by a blank.
390(186)	8	EBCDIC	2	Primary (client) user ID for this nested ACEE.
391(187)	8	EBCDIC	69, 70, 72, 73, 74, 80, 83, 85, 89	Domain name of the target PKI Services certificate authority.
392(188)	1-510	EBCDIC	All, except 68, 71, 79, 81, 82, and 85	Authenticated user name.
393(189)	1-255	EBCDIC	All, except 68, 71, 79, 81, 82, and 85	Authenticated user registry name.
394(18A)	1-128	EBCDIC	All, except 68, 71, 79, 81, 82, and 85	Authenticated user host name.
395(18B)	1-16	EBCDIC	All, except 68, 71, 79, 81, 82, and 85	Authenticated user authentication mechanism object identifier (OID).
396(18C)	3-244	EBCDIC	2	Access criteria. Note: When this relocate is used, the data appears in the form of <i>criteria-name=criteria-value</i> .
398(18E)	1-64	EBCDIC	66	PKDS label.
399(18F)	1-32	EBCDIC	66	Token name.
400(190)	8	EBCDIC	84	Ring owner.
401(191)	1	Binary	84	Reuse attribute flag for NewRing.
402(192)	1	Binary	84	Trust attribute flag for DataPut.
403(193)	1	Binary	84	HighTrust attribute flag for DataPut.
404(194)	1	Binary	84	Delete attribute flag for DataRemove.
405(195)	8	EBCDIC	84	Certificate usage: 'SITE', 'CERTAUTH' or 'PERSONAL'.
406(196)	1	Binary	84	Default flag. X'01' means default certificate.
407(197)	1	Binary	84	Private key specified. X'01' means that a private key is specified.
408(198)	256	EBCDIC	85	AutoRenew Exit path name.

Table 3. Table of extended-length relocate section variable data (continued)				
Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
409(199)	1-255	EBCDIC	86	Root signing certificate subject's distinguished name
410(19A)	1-255	EBCDIC	86	Program signer (end entity) certificate subject's distinguished name
411(19B)	1	Binary	86	R_PgmSignVer flags byte Bit Meaning 0 1 = Module allowed to be loaded
412(19C)	8	EBCDIC	86	Time module was signed
413(19D)	10	EBCDIC	86	Date module was signed
414(19E)	10	EBCDIC	86	Date when module certificate chain expires
415(19F)	1-246	EBCDIC	87	Value of the user ID filter from the USERIDFILTER keyword on MAP
416(1A0)	1-255	EBCDIC	87	Value of the registry name from the REGISTRY keyword of RACMAP
417(1A1)	1-20	EBCDIC	88	Service or process name for automatically updated profile
418(1A2)	1-8	EBCDIC	88	Class for automatically updated profile
419(1A3)	1-255	EBCDIC	88	Automatically updated profile name
420(1A4)	1-4000	EBCDIC	88	Automatically updated profile data
421(1A5)	40	EBCDIC	70, 72, 89	Key ID
422(1A6)	4	EBCDIC	69	Key size
423(1A7)	32	EBCDIC	74	Requester email
424(1A8)	1-246	UTF-8	All, except 68, 71, 79, 81, 82, and 85	Authenticated distributed-identity user name
425(1A9)	1-246	UTF-8	All, except 68, 71, 79, 81, 82, and 85	Authenticated distributed-identity registry name
426(1AA)	10	EBCDIC	69	Key algorithm
427(1AB)	1024	EBCDIC	69, 73, 83	Customized extension
428(1AC)	32	EBCDIC	69, 73, 83	Record link
429(1AD)	32	EBCDIC	72	Signing Algorithm
430(1AE)				Reserved
431(1AF)				Reserved
432(1B0)				Reserved
433(1B1)	2	Unsigned	72	Number of approvals required for the request
434(1B2)	2	Unsigned	72	Count of approvals performed
435(1B3)	1	Binary	84	Notrust attribute flag for DataPut and DataAlter
436(1B4)	1	Binary	84	Delete attribute flag for DataRemove, even if the certificate is connected to rings
437(1B5)	1	Binary	84	Delete attribute flag for DataRemove, even if the certificate is used for GENREQ
438(1B6)	32	EBCDIC	84	Source certificate label

Table 3. Table of extended-length relocate section variable data (continued)

Table 3. Table of extended-length relocate section variable data (continued)																																
Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)																												
440(1B8)	8	binary	13	<p>Byte 1: MFA subkeyword specified flags</p> <table><thead><tr><th>Bit</th><th>Meaning</th></tr></thead><tbody><tr><td>0</td><td>PWFALLBACK specified</td></tr><tr><td>1</td><td>NOPWFALLBACK specified</td></tr><tr><td>2</td><td>FACTOR specified</td></tr><tr><td>3</td><td>DELFACOR specified</td></tr><tr><td>4</td><td>ACTIVE specified</td></tr><tr><td>5</td><td>NOACTIVE specified</td></tr><tr><td>6</td><td>TAGS specified</td></tr><tr><td>7</td><td>DELTAGS specified</td></tr></tbody></table> <p>Byte 2: MFA subkeyword specified flags</p> <table><thead><tr><th>Bit</th><th>Meaning</th></tr></thead><tbody><tr><td>0</td><td>NOTAGS specified</td></tr><tr><td>1</td><td>ADDPOLICY specified</td></tr><tr><td>2</td><td>DELPOLICY specified</td></tr><tr><td>3-7</td><td>Reserved</td></tr></tbody></table> <p>Bytes 3-8: Reserved for IBM's use</p>	Bit	Meaning	0	PWFALLBACK specified	1	NOPWFALLBACK specified	2	FACTOR specified	3	DELFACOR specified	4	ACTIVE specified	5	NOACTIVE specified	6	TAGS specified	7	DELTAGS specified	Bit	Meaning	0	NOTAGS specified	1	ADDPOLICY specified	2	DELPOLICY specified	3-7	Reserved
Bit	Meaning																															
0	PWFALLBACK specified																															
1	NOPWFALLBACK specified																															
2	FACTOR specified																															
3	DELFACOR specified																															
4	ACTIVE specified																															
5	NOACTIVE specified																															
6	TAGS specified																															
7	DELTAGS specified																															
Bit	Meaning																															
0	NOTAGS specified																															
1	ADDPOLICY specified																															
2	DELPOLICY specified																															
3-7	Reserved																															
441(1B9)	variable	EBCDIC	13	Multifactor authentication factor name																												
442(1BA)	variable	EBCDIC	13	<p>MFA tag entry from the TAGS/DELTAGS keyword.</p> <p>When TAGS is specified, the entry value is the tag name and value separated by a colon (":"). When DELTAGS is specified, the entry value is the tag name only.</p>																												

Table 3. Table of extended-length relocate section variable data (continued)

Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
443(1BB)	variable	mixed	1	<p>Byte 1: Authentication information:</p> <p>Bit Meaning</p> <p>0 ACEE was created from VLF cache</p> <p>1 User has active MFA factor(s)</p> <p>2 MFA user allowed to fall back when no MFA decision can be made</p> <p>3 No MFA decision for MFA user</p> <p>4 IBM MFA requested that RACROUTE REQUEST=VERIFY return the password-expired return code.</p> <p>5 IBM MFA requested that RACROUTE REQUEST=VERIFY return the new-password-invalid return code.</p> <p>6 IBM MFA requested that RACROUTE REQUEST=VERIFY return the password-invalid return code, but not to increment the password revoke count (partial success - needs more information).</p> <p>7 Relocate 443 is extended.</p> <p>Byte 2: Authenticators used:</p> <p>Bit Meaning</p> <p>0 Password Evaluated</p> <p>1 Password Successful</p> <p>2 Password Phrase Evaluated</p> <p>3 Password Phrase Successful</p> <p>4 Passticket Evaluated</p> <p>5 Passticket Successful</p> <p>6 MFA authentication successful</p> <p>7 MFA authentication unsuccessful</p> <p>Bytes 3-6: MFA Authorization Return Code.</p> <p>Bytes 7-10: MFA Authorization Reason Code</p>

Table 3. Table of extended-length relocate section variable data (continued)

Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
443(1BB) (Cont.)	variable	mixed	1	<p>Note: Below fields are present only when relocate 443 is extended.</p> <p>Bytes 11-14: PassTicket Return Code</p> <p>Bytes 15-18: PassTicket Reason Code</p> <p>Byte 19: Flag byte 3: Authentication Details</p> <p>Bit</p> <p>Meaning when set</p> <p>0 Password or Password Phrase expired</p> <p>1 New Password or Password Phrase invalid</p> <p>2 Identity Token (IDT) Evaluated</p> <p>3 Identity Token (IDT) Successful</p> <p>4 IBM MFA requested that RACROUTE REQUEST=VERIFY return the password-invalid return code, but not to increment the password revoke count (reauthentication requested).</p> <p>5 Legacy PassTicket Evaluated</p> <p>6 Legacy PassTicket Successful</p> <p>7 Enhanced PassTicket Type UPPER Evaluated</p> <p>Byte 20: Flag byte 4: Authentication Details</p> <p>Bit</p> <p>Meaning when set</p> <p>0 Enhanced PassTicket Type UPPER Successful</p> <p>1 Enhanced PassTicket Type MIXED Evaluated</p> <p>2 Enhanced PassTicket Type MIXED Successful</p> <p>3 IDT from existing security environment.</p> <p>4 Relocate 443 is extended part 2.</p> <p>5-7 Reserved</p> <p>Bytes 21-28: Derived Application Name</p> <p>Bytes 29-32: IDT Validation Reason Code</p> <p>Bytes 33-36: IDT Error Reason Code</p> <p>Bytes 37-40: Failing Service ID</p> <p>Bytes 41-44: Failing Service Return Code</p> <p>Bytes 45-48: Failing Service Reason Code</p>
443(1BB) (Cont.)	variable	mixed	1	<p>Note: The following fields are present only when relocate 443 is extended part 2.</p> <p>Bytes 49-58: Signature algorithm from the provided IDT</p> <p>Bytes 59-90: Key identifier from the provided IDT</p> <p>Bytes 91-190: Reserved.</p> <p>Bytes 191-290: Reserved.</p> <p>Bytes 291-536: Reserved.</p> <p>>>>>>> origin/Draft</p>

Table 3. Table of extended-length relocate section variable data (continued)				
Data type (SMF80TP2) dec(hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
443(1BB) (Cont.)	variable	mixed	1	<p>Note: The following fields are present only when relocate 443 is extended part 2.</p> <p>Bytes 49-58: Signature algorithm from the provided IDT</p> <p>Bytes 59-90: Key identifier from the provided IDT</p> <p>Bytes 91-190: Reserved.</p> <p>Bytes 191-290: Reserved.</p> <p>Bytes 291-536: Reserved.</p> <p>Byte 537: Flag byte 5: Authentication Details.</p> <p>Bit</p> <p>Meaning when set</p> <p>0</p> <p>IDT signature evaluated with primary label</p> <p>1</p> <p>IDT signature evaluated with token</p> <p>2-7</p> <p>Reserved.</p>
444 (1BC)	variable	EBCDIC	13	MFA policy name entry from the ADDPOLICY/DELPOLICY keyword.
445 (1BD)	variable	mixed	2	<p>Bytes</p> <p>Meaning</p> <p>1 - 2</p> <ul style="list-style-type: none"> Includes the length of the LOGSTRX ID and length field Maximum length value is 1100 bytes <p>3 - 4</p> <ul style="list-style-type: none"> IDs from X'0000' to X'0FFF' are reserved for IBM <ul style="list-style-type: none"> X'0001' identifies the data as a CICS identity IDs from X'1000' to X'1FFF' are reserved for vendors IDs from X'2000' to X'FFFF' are reserved for customer data. <p>5 - end</p> <p>Triplet of values per field each consisting of:</p> <ul style="list-style-type: none"> 2 byte ID values 2 byte length Variable data <p>Note: For more information about REQUEST=FASTAUTH, see z/OS Security Server RACROUTE Macro Reference.</p>
446 (1BE)	32	unsigned	66, 67, 69, 70, 74, 83, 84, 85, 90	Subject Certificate fingerprint
447 (1BF)	32	unsigned	85, 90	Issuer Certificate fingerprint
448 (1C0)	32	unsigned	85	Previous Certificate fingerprint
449 (1C1)	variable	mixed	67	<p>Byte 1-8: User ID</p> <p>Byte 9-16: Application name</p> <p>Byte 17-20: IDT Build Reason Code</p> <p>Byte 21-24: Failing Service Identifier</p> <p>Byte 25-28: Failing Service Return Code</p> <p>Byte 29-32: Failing Service Reason Code</p>

Table of data type 6 command-related data

This table describes the RACF command-related data that is associated with data type 6.

- ADDGROUP
- ADDSD

- ADDUSER
- ALTDSD
- ALTGROUP
- ALTUSER
- CONNECT
- DELDSD
- DELGROUP
- DELUSER
- PASSWORD
- PERMIT
- RACDCERT
- RACLINK
- RACMAP
- RALTER
- RDEFINE
- RDELETE
- REMOVE
- RVARY
- SETROPTS

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

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

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

The 'flags for keywords ignored due to error conditions' field indicates individual keywords that were not processed for reasons other than insufficient authority. Not all commands (event codes 8-25) have these flags. The keyword errors are not terminating errors (like the errors that are indicated in SMF80ERR) and the command continues processing other specified operands. If a terminating error, these flags do not necessarily indicate what processing was done or not done. Any keyword errors occurring before the terminating error are indicated, but the keywords, that are not processed because of a terminating error, are not indicated. The bits in SMF80ERR indicate whether RACF already made changes to the RACF database before the terminating error or whether no updates were made.

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

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

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


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

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

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

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

Event code dec(hex)	Command	Data length	Format	Description
8(8)	ADDSD	2	Binary	<p>Flags for keywords specified:</p> <p>Bit - Keyword specified</p> <p>Byte 0</p> <p>Bit</p> <p>Keyword specified</p> <p>Byte 0</p> <p>0 VOLUME</p> <p>1 UNIT</p> <p>2 UACC</p> <p>3 OWNER</p> <p>4 AUDIT</p> <p>5 SET</p> <p>6 NOSET</p> <p>7 LEVEL</p> <p>Byte 1</p> <p>0 PASSWORD</p> <p>1 DATA</p> <p>2 MODEL</p> <p>3 WARNING</p> <p>4 GENERIC</p> <p>5 SECLEVEL</p> <p>6 ADDCATEGORY</p> <p>7 NOTIFY</p>
		2	Binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
8(8) (Cont.)	ADDSD (Cont.)	44	EBCDIC	Data set name
		8	EBCDIC	Type (UNIT keyword)
		1	Binary	Flags for UACC keyword: Note: If this is a non-DFP data set, RACF ignores bit 4 when checking access to data sets. Bit Authority specified 0 ALTER 1 CONTROL 2 UPDATE 3 READ 4 EXECUTE 5–6 Reserved for IBM's use 7 NONE
		8	EBCDIC	User ID or group name (OWNER keyword)
		1	Binary	Flags for AUDIT keyword: (only one set at a time) Bit Option specified 0 ALL 1 SUCCESS 2 FAILURES 3 NONE 4–5 SUCCESS qualifier codes: ‘00’ READ ‘01’ UPDATE ‘10’ CONTROL ‘11’ ALTER 6–7 FAILURES qualifier codes: ‘00’ READ ‘01’ UPDATE ‘10’ CONTROL ‘11’ ALTER
		1	Binary	nn (LEVEL keyword)
		1	Binary	Flags for RACF processing: Bit Meaning 0 Data set profile inconsistent with RACF indicator 1 Generic profile name specified 2 FROM entity is longer than 44 characters entity is passed in relocate type 13 3–7 Reserved for IBM's use
		8	EBCDIC	User to be notified when this profile denies access

Event code dec(hex)	Command	Data length	Format	Description
		2	Binary	Flags for keywords specified: Bit Keyword specified Byte 0 0 SETONLY 1 TAPE 2 FILESEQ 3 RETPD 4 ERASE 5 FROM 6 FCLASS 7 FVOLUME Byte 1 0 FGENERIC 1 SECLABEL 2–7 Reserved for IBM's use
		2	Binary	Flags for keywords ignored. Same format as flags for keywords specified.
		1	EBCDIC	Reserved for IBM's use
		2	Binary	File sequence number
		2	Binary	Retention period
		8	EBCDIC	FROM class name
		44	EBCDIC	FROM resource name
		8	EBCDIC	FROM volume serial
		44	EBCDIC	SECLEVEL name
		8	EBCDIC	SECLABEL

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
9(9)	ADDGROUP	1	Binary	Flags for keywords specified:
				Bit
				Keyword specified
				0 SUPGROUP
				1 OWNER
				2 NOTERMUACC
				3 TERMUACC
				4 DATA
				5 MODEL
				6 UNIVERSAL
				7 Reserved for IBM's use
		1	Binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		8	EBCDIC	Group name
		8	EBCDIC	Superior group name (SUPGROUP keyword)
		8	EBCDIC	User ID or group name (OWNER keyword)

Event code dec(hex)	Command	Data length	Format	Description
10(A)	ADDUSER	* The data for event code 10 is identical to the data for event code 13, with these exceptions.		
		4	Binary	Flags for keywords specified:
				Bit
				Keyword specified
				Byte 0
				0
				DFLTGRP
				*1
				GROUP
				2
				PASSWORD
				3
				NOPASSWORD
				4
				NAME
				5
				AUTHORITY
				6
				DATA
				7
				GRPACC
				Byte 1
				0
				NOGRPACC
				1
				UACC
				2
				ADSP
				3
				NOADSP
				4
				OWNER
				5
				SPECIAL
				6
				NOSPECIAL
				7
				OPERATIONS
				Byte 2
				0
				NOOPERATIONS
				1
				CLAUTH
				2
				NOCLAUTH
				3
				AUDITOR
				4
				NOAUDITOR
				5
				OIDCARD
				6
				NOOIDCARD
				*7
				REVOKE

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
10(A) (Cont.)	ADDUSER (Cont.)	4	Binary	Byte 3
				*0 RESUME
				*1 AUDIT
				*2 NOAUDIT
				3 MODEL
				*4 NOMODEL
				5 WHEN
				6 ADDCATEGORY
				7 DELCATEGORY
		4	Binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		4	Binary	Flags for keywords ignored because of error conditions
		1	Binary	Flags for other violations:
				Bit Violation
				*0 Command invoker does not have CLAUTH attribute of USER
				1 Command invoker does not have sufficient authority to group
				*2 Command invoker does not have sufficient authority to user profile
				*3–7 Reserved for IBM's use
		8	EBCDIC	User ID
		8	EBCDIC	Group name (DFLTGRP keyword)
		8	EBCDIC	*Group name (GROUP keyword)
10(A) (Cont.)	ADDUSER (Cont.)	1	Binary	Flags for AUTHORITY keyword:
				Bit Authority specified
				0 JOIN
				1 CONNECT
				2 CREATE
				3 USE
				4–7 Reserved for IBM's use

Event code dec(hex)	Command	Data length	Format	Description
		1	Binary	Flags for UACC keyword: Bit Authority specified 0 ALTER 1 CONTROL 2 UPDATE 3 READ 4–6 Reserved for IBM's use 7 NONE
		8	EBCDIC	User ID or group name (OWNER keyword)
		2	Binary	Flags for classes specified (CLAUTH keyword) Bit Keyword specified Byte 0 0–1 Reserved for IBM's use 2 USER 3 Reserved for IBM's use 4 DASDVOL 5 TAPEVOL 6 TERMINAL 7 Reserved for IBM's use Byte 1 0–7 Reserved for IBM's use
		2	Binary	Flags for classes ignored because of insufficient authority: Same format as flags for classes specified. Note: if all classes specified are ignored because of insufficient authority, then the 'flags for keywords ignored because of insufficient authority' field indicates that CLAUTH was ignored.

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
		2	Binary	Flags for additional keywords specified: Bit Keyword specified Byte 0 0 SECLEVEL 1 NOSECLEVEL 2 SECLABEL 3 NOSECLABEL 4 NOEXPIRED 5 EXPIRED 6 RESTRICTED 7 NORESTRICTED Byte 1 0 Reserved for IBM's use 1 Reserved for IBM's use 2 PHRASE 3 NOPHRASE 4-5 Reserved for IBM's use 6 ROAUDIT 7 NOROAUDIT

Event code dec(hex)	Command	Data length	Format	Description
10(A) (Cont.)	ADDUSER (Cont.)	2	Binary	<p>Flags for additional keywords ignored (authorization):</p> <p>Bit</p> <p>Keyword ignored</p> <p>Byte 0</p> <p>0 SECLEVEL</p> <p>1 NOSECLEVEL</p> <p>2 SECLABEL</p> <p>3 NOSECLABEL</p> <p>4 NOEXPIRED</p> <p>5 EXPIRED</p> <p>6 RESTRICTED</p> <p>7 NORESTRICTED</p> <p>Byte 1</p> <p>0 Reserved for IBM's use</p> <p>1 Reserved for IBM's use</p> <p>2 PHRASE</p> <p>3 NOPHRASE</p> <p>4-5 Reserved for IBM's use</p> <p>6 ROAUDIT</p> <p>7 NOROAUDIT</p>

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
		2	Binary	<p>Flags for additional keywords ignored because of processing error:</p> <p>Bit</p> <p>Keyword specified</p> <p>Byte 0</p> <p>0 SECLEVEL</p> <p>1 NOSECLEVEL</p> <p>2 SECLABEL</p> <p>3 NOSECLABEL</p> <p>4 Reserved for IBM's use</p> <p>5 Reserved for IBM's use</p> <p>6 RESTRICTED</p> <p>7 NORESTRICTED</p> <p>Byte 1</p> <p>0-4 Reserved for IBM's use</p> <p>5 ROAUDIT</p> <p>6 NOROAUDIT</p> <p>7 Reserved for IBM's use</p>
		3	packed	Logon time (packed); if time is not specified, this field contains binary zeros; if TIME(ANYTIME) is specified, this field contains X'F0F0F0'.
		3	packed	Logoff time (packed); if time is not specified, this field contains binary zeros; if TIME(ANYTIME) is specified, this field contains X'F0F0F0'.
10(A) (Cont.)	ADDUSER (Cont.)	1	Binary	<p>Logon day</p> <p>Bit</p> <p>Days the user cannot log on</p> <p>0 Sunday</p> <p>1 Monday</p> <p>2 Tuesday</p> <p>3 Wednesday</p> <p>4 Thursday</p> <p>5 Friday</p> <p>6 Saturday</p> <p>7 Day not specified</p>
		4	EBCDIC	REVOKE date
		4	EBCDIC	RESUME date
		44	EBCDIC	SECLEVEL name
		8	EBCDIC	SECLABEL name

Event code dec(hex)	Command	Data length	Format	Description
11(B)	ALTDSD	2	Binary	<p>Flags for keywords specified:</p> <p>Bit</p> <p>Keyword specified</p> <p>Byte 0</p> <p>0 OWNER</p> <p>1 UACC</p> <p>2 AUDIT</p> <p>3 LEVEL</p> <p>4 ADDVOL</p> <p>5 DELVOL</p> <p>6 SET</p> <p>7 NOSET</p> <p>Byte 1</p> <p>0 GLOBALAUDIT</p> <p>1 VOLUME</p> <p>2 PASSWORD</p> <p>3 UNIT</p> <p>4 ALTVOL</p> <p>5 DATA</p> <p>6–7 Reserved for IBM's use</p>
		2	Binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified, except that Byte 1, Bit 2 is reserved for IBM's use.
		2	Binary	Flags for keywords ignored because of error conditions: Same format as flags for keywords specified, except that Byte 1, Bit 2 is reserved for IBM's use.
		44	EBCDIC	Data set name
		8	EBCDIC	User ID or group name (OWNER keyword)

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
		1	Binary	Flags for UACC keyword: Note: If this is a non-DFP data set, RACF ignores bit 4 when checking access to the data set. Bit Authority specified 0 ALTER 1 CONTROL 2 UPDATE 3 READ 4 EXECUTE 5–6 Reserved for IBM's use 7 NONE
		1	Binary	Flags for AUDIT keyword: Bit Option specified 0 ALL 1 SUCCESS 2 FAILURES 3 NONE 4–5 SUCCESS qualifier codes 6–7 FAILURES qualifier codes
		1	Binary	nn (LEVEL keyword)
		1	Binary	Flags for GLOBALAUDIT keyword: Same format as flags for AUDIT keyword.
		6	EBCDIC	Volume serial ID (VOLUME keyword)
11(B) (Cont.)	ALTDSD (Cont.)	8	EBCDIC	Unit information
		1	Binary	Flags for RACF processing: Bit Meaning 0 Profile inconsistent with RACF indicator. 1 Generic profile name specified 2–7 Reserved for IBM's use

Event code dec(hex)	Command	Data length	Format	Description
		2	Binary	<p>Additional keywords specified:</p> <p>Bit</p> <p>Keyword specified</p> <p>Byte 0</p> <p>0 GENERIC</p> <p>1 WARNING</p> <p>2 NOWARNING</p> <p>3 ERASE</p> <p>4 NOERASE</p> <p>5 RETPD</p> <p>6 NOTIFY</p> <p>7 NONOTIFY</p> <p>Byte 1</p> <p>0 SECLEVEL</p> <p>1 ADDCATEGORY</p> <p>2 DELCATEGORY</p> <p>3 NOSECLEVEL</p> <p>4 SECLABEL</p> <p>5 NOSECLABEL</p> <p>6–7 Reserved for IBM's use</p>
		2	Binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		2	Binary	Flags for keywords ignored because of a processing error: Same format as flags for keywords specified.
		2	Binary	Retention period
		8	EBCDIC	User to be notified when access denied.
		44	EBCDIC	SECLEVEL name
		8	EBCDIC	SECLABEL name
12(C)	ALTGROUP	1	Binary	<p>Flags for keywords specified:</p> <p>Bit</p> <p>Keyword specified</p> <p>0 SUPGROUP</p> <p>1 OWNER</p> <p>2 NOTERMUACC</p> <p>3 TERMUACC</p> <p>4 DATA</p> <p>5 MODEL</p> <p>6–7 Reserved for IBM's use</p>

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
		1	Binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keyword's specified.
		1	Binary	Flags for other violations: Bit Violation 0 Lack of proper authority to old SUPGROUP 1–7 Reserved for IBM's use
		8	EBCDIC	Group name
		8	EBCDIC	Superior group name (SUPGROUP keyword)
		8	EBCDIC	User ID or group name (OWNER keyword)
		1	Binary	Flags for keywords ignored because of error conditions: Same format as flags for keywords specified.
13(D)	ALTUSER	* The data for event code 13 is identical to the data for event code 10, with these exceptions.		
		4	Binary	Flags for keywords specified: Bit Keyword specified Byte 0 0 DFLTGRP *1 GROUP 2 PASSWORD 3 NOPASSWORD 4 NAME 5 AUTHORITY 6 DATA 7 GRPACC Byte 1 0 NOGRPACC 1 UACC 2 ADSP 3 NOADSP 4 OWNER 5 SPECIAL 6 NOSPECIAL 7 OPERATIONS

Event code dec(hex)	Command	Data length	Format	Description
		4	Binary	Byte 2 0 NOOPERATIONS 1 CLAUTH 2 NOCLAUTH 3 AUDITOR 4 NOAUDITOR 5 OIDCARD 6 NOOIDCARD *7 REVOKE Byte 3 *0 RESUME *1 UAUDIT *2 NOUAUDIT 3 MODEL 4 NOMODEL 5 WHEN 6 ADDCATEGORY 7 DELCATEGORY
		4	Binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		4	Binary	Flags for keywords ignored because of error conditions: Same format as flags for keywords specified.
		1	Binary	Flags for other violations: Bit Violation *0 Command invoker does not have CLAUTH attribute of USER 1 Command invoker does not have sufficient authority to group *2 Command invoker does not have sufficient authority to user profile 3 Reserved for IBM's use 4 NOEXPIRED 5 EXPIRED 6-7 Reserved for IBM's use
13(D) (Cont.)	ALTUSER (Cont.)	8	EBCDIC	User ID
		8	EBCDIC	Group name (DFLTGRP keyword)
		8	EBCDIC	*Group name (GROUP keyword)

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
		1	Binary	Flags for AUTHORITY keyword: Bit Authority specified 0 JOIN 1 CONNECT 2 CREATE 3 USE 4–7 Reserved for IBM's use
		1	Binary	Flags for UACC keyword: Bit Authority specified 0 ALTER 1 CONTROL 2 UPDATE 3 READ 4–6 Reserved for IBM's use 7 NONE
		8	EBCDIC	User ID (OWNER keyword)
		2	Binary	Flags for classes specified (CLAUTH keywords) Bit Option specified Byte 0 0–1 Reserved for IBM's use 2 USER 3 Reserved for IBM's use 4 DASDVOL 5 TAPEVOL 6 TERMINAL 7 Reserved for IBM's use Byte 1 0–7 Reserved for IBM's use
		2	Binary	Flags for classes ignored because of insufficient authority: Same format as flags for classes specified. Note that if all classes specified are ignored because of insufficient authority, then the 'flags for keywords ignored because of insufficient authority' field indicates that CLAUTH or NOCLAUTH was ignored.

Event code dec(hex)	Command	Data length	Format	Description
		2	Binary	Flags for additional keywords specified: Bit Keyword specified Byte 0 0 SECLEVEL *1 NOSECLEVEL *2 SECLABEL *3 NOSECLABEL *4 NOEXPIRED *5 EXPIRED *6 RESTRICTED *7 NORESTRICTED
13(D) (Cont.)	ALTUSER (Cont.)	2	Binary	Flags for additional keywords specified: Byte 1 0 NOREVOKE 1 NORESUME 2 PHRASE 3 NOPHRASE 4 *PWCLEAN 5 *PWCONVERT 6 ROAUDIT 7 NOROAUDIT

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
		2	Binary	Flags for additional keywords ignored (authorization): Bit Keyword ignored Byte 0 0 SECLEVEL *1 NOSECLEVEL *2 SECLABEL *3 NOSECLABEL *4 NOEXPIRED *5 EXPIRED *6 RESTRICTED *7 NORESTRICTED Byte 1 0 NOREVOKE 1 NORESUME 2 PHRASE 3 NOPHRASE 4 *PWCLEAN 5 *PWCONVERT 6 ROAUDIT 7 NOROAUDIT

Event code dec(hex)	Command	Data length	Format	Description
		2	Binary	<p>Flags for additional keywords ignored because of processing error:</p> <p>Bit</p> <p>Keyword specified</p> <p>Byte 0</p> <p>0 SECLEVEL</p> <p>*1 NOSECLEVEL</p> <p>*2 SECLABEL</p> <p>*3 NOSECLABEL</p> <p>*4 NOEXPIRED</p> <p>*5 EXPIRED</p> <p>*6 RESTRICTED</p> <p>*7 NORESTRICTED</p> <p>Byte 1</p> <p>0 *PWCLEAN</p> <p>1 *PWCONVERT</p> <p>2-4 Reserved for IBM's use</p> <p>5 ROAUDIT</p> <p>6 NOROAUDIT</p> <p>7 Reserved for IBM's use</p>
		3	packed	Logon time (packed); if time is not specified, this field contains binary zeros; if TIME(ANYTIME) is specified, this field contains X'F0F0F0'.
		3	packed	Logoff time (packed); if time is not specified, this field contains binary zeros; if TIME(ANYTIME) is specified, this field contains X'F0F0F0'.
13(D) (Cont.)	ALTUSER (Cont.)	1	Binary	<p>Days the user cannot log on</p> <p>Bit</p> <p>Day specified</p> <p>0 Sunday</p> <p>1 Monday</p> <p>2 Tuesday</p> <p>3 Wednesday</p> <p>4 Thursday</p> <p>5 Friday</p> <p>6 Saturday</p> <p>7 Day not specified</p>
		4	EBCDIC	REVOKE date
		4	EBCDIC	RESUME date
		44	EBCDIC	SECLEVEL name
		8	EBCDIC	SECLABEL name

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
13 (D)	ALTUSER	4	Binary	<p>Flags for additional keywords specified:</p> <p>Bit Keyword specified</p> <p>Byte 0</p> <p>0 *MFA</p> <p>1 *NOMFA</p> <p>2-7 Reserved for IBM's use</p> <p>Byte 1</p> <p>0-7 Reserved for IBM's use</p> <p>Byte 2</p> <p>0-7 Reserved for IBM's use</p> <p>Byte 3</p> <p>0-7 Reserved for IBM's use</p>
		4	Binary	<p>Flags for additional keywords ignored (authorization):</p> <p>Bit Keyword specified</p> <p>Byte 0</p> <p>0 *MFA</p> <p>1 *NOMFA</p> <p>2-7 Reserved for IBM's use</p> <p>Byte 1</p> <p>0-7 Reserved for IBM's use</p> <p>Byte 2</p> <p>0-7 Reserved for IBM's use</p> <p>Byte 3</p> <p>0-7 Reserved for IBM's use</p>
		4	Binary	<p>Flags for additional keywords ignored because of processing error:</p> <p>Bit Keyword specified</p> <p>Byte 0</p> <p>0 *MFA</p> <p>1 *NOMFA</p> <p>2-7 Reserved for IBM's use</p> <p>Byte 1</p> <p>0-7 Reserved for IBM's use</p> <p>Byte 2</p> <p>0-7 Reserved for IBM's use</p> <p>Byte 3</p> <p>0-7 Reserved for IBM's use</p>

Event code dec(hex)	Command	Data length	Format	Description
14(E)	CONNECT	2	Binary	<p>Flags for keywords specified:</p> <p>Bit</p> <p>Keyword specified</p> <p>Byte 0</p> <p>0 GROUP</p> <p>1 UACC</p> <p>2 AUTHORITY</p> <p>3 ADSP</p> <p>4 NOADSP</p> <p>5 REVOKE</p> <p>6 RESUME</p> <p>7 GRPACC</p> <p>Byte 1</p> <p>0 NOGRPACC</p> <p>1 OPERATIONS</p> <p>2 NOOPERATIONS</p> <p>3 SPECIAL</p> <p>4 NOSPECIAL</p> <p>5 AUDITOR</p> <p>6 NOAUDITOR</p> <p>7 OWNER</p>
		2	Binary	<p>Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.</p>

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
14(E) (Cont.)	CONNECT (Cont.)	8	EBCDIC	User ID
		8	EBCDIC	Group name (GROUP keyword)
		1	Binary	Flags for UACC keyword: Bit Authority specified 0 ALTER 1 CONTROL 2 UPDATE 3 READ 4–6 Reserved for IBM's use 7 NONE
		1	Binary	Flags for AUTHORITY keyword: Bit Authority specified 0 JOIN 1 CONNECT 2 CREATE 3 USE 4–7 Reserved for IBM's use
		1	Binary	Flags for additional keywords specified Bit Keyword specified 0 NOREVOKE 1 NORESUME 2–7 Reserved for IBM's use
		1	Binary	Flags for additional keywords ignored because of insufficient authority. Same format as flags for additional keywords specified.
		8	EBCDIC	User ID or group name (OWNER keyword)
		4	packed	REVOKE date, packed
		4	packed	RESUME date, packed

Event code dec(hex)	Command	Data length	Format	Description
15(F)	DELDSD	1	Binary	Flags for keywords specified or taken as defaults: Bit Keyword specified 0 SET 1 NOSET 2 VOLUME 3 GENERIC 4–7 Reserved for IBM's use
		1	Binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		44	EBCDIC	Data set name
		6	EBCDIC	Volume serial ID (VOLUME keyword)
		1	Binary	Flags for RACF processing: Bit Meaning 0 Profile inconsistent with RACF indicator 1 Generic profile name specified 2–7 Reserved for IBM's use
16(10)	DELGROUP	8	EBCDIC	Group name
17(11)	DELUSER	8	EBCDIC	User ID
18(12)	PASSWORD	1	Binary	Flags for keywords specified: Bit Keyword specified 0 INTERVAL 1 USER 2 PASSWORD 3 PHRASE 4 PHRASEINT 5–7 Reserved for IBM's use
		1	Binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		1	Binary	Flags for keywords ignored because of error conditions: Same format as flags for keywords specified.
		4	Binary	Change-interval (INTERVAL keyword) Note: If the NOINTERVAL keyword is specified, the change-interval changes to X'FF'.
		8	EBCDIC	User ID (USER keyword)
		4	Binary	Password phrase change-interval (PHRASEINT keyword) Note: If the NOPHRASEINT keyword is specified, the Password phrase change-interval changes to 65535 (X'FFFF').

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
19(13)	PERMIT	2	Binary	<p>Flags for keywords specified or taken as defaults:</p> <p>Bit</p> <p>Keyword specified</p> <p>Byte 0</p> <p>0 CLASS</p> <p>1 ID</p> <p>2 ACCESS</p> <p>3 FROM</p> <p>4 DELETE</p> <p>5 FCLASS</p> <p>6 VOLUME</p> <p>7 FVOLUME</p> <p>Byte 1</p> <p>0 GENERIC</p> <p>1 FGENERIC</p> <p>2 RESET</p> <p>3 WHEN</p> <p>4 RESET(WHEN)</p> <p>5 RESET(STANDARD)</p> <p>6–7 Reserved for IBM's use</p>
		2	Binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified, except that bits are not set for RESET(STANDARD) or RESET(WHEN).
		2	Binary	Flags for keywords ignored because of error conditions: Same format as flags for keywords specified, except that bits are not set for RESET(STANDARD) or RESET(WHEN).

Event code dec(hex)	Command	Data length	Format	Description
		2	Binary	<p>Flags for CLASS keyword, and for the RESET keyword:</p> <p>Bit</p> <p>Option specified</p> <p>Byte 0</p> <p>0–2 Reserved for IBM's use</p> <p>3 DATASET</p> <p>4 DASDVOL</p> <p>5 TAPEVOL</p> <p>6 TERMINAL</p> <p>7 Reserved for IBM's use</p> <p>Byte 1</p> <p>0 FROM generic resource</p> <p>1–5 Reserved for IBM's use</p> <p>6 Conditional access list is indicated by RESET keyword.</p> <p>7 Standard access list is indicated by RESET keyword.</p>
19(13) (Cont.)	PERMIT (Cont.)	1	Binary	<p>Flags for ACCESS keyword:</p> <p>Note: If this is a non-DFP data set, RACF ignores bit 4 when checking access to the data set.</p> <p>Bit</p> <p>Authority specified</p> <p>0 ALTER</p> <p>1 CONTROL</p> <p>2 UPDATE</p> <p>3 READ</p> <p>4 EXECUTE</p> <p>5–6 Reserved for IBM's use</p> <p>7 NONE</p>
		2	Binary	<p>Flags for FCLASS keyword:</p> <p>Same format as flags for CLASS keyword.</p>
20(14)	RALTER	* The data for event code 20 is identical to the data for event code 21, with these exceptions.		

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
		2	Binary	<p>Flags for keywords specified:</p> <p>Bit Keyword specified</p> <p>Byte 0</p> <p>0 DATA</p> <p>1 OWNER</p> <p>2 UACC</p> <p>3 LEVEL</p> <p>4 AUDIT</p> <p>*5 GLOBALAUDIT</p> <p>*6 ADDVOL</p> <p>*7 DELVOL</p> <p>Byte 1</p> <p>0 ADDMEM</p> <p>1 DELMEM</p> <p>2 APPLDATA</p> <p>3 SINGLEDSN</p> <p>*4 NOSINGLEDSN</p> <p>5 WARNING</p> <p>6 NOWARNING</p> <p>7 WHEN</p>
		2	Binary	<p>Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.</p>
		2	Binary	<p>Flags for class name:</p> <p>Bit Option specified</p> <p>Byte 0</p> <p>0–3 Reserved for IBM's use</p> <p>4 DASDVOL</p> <p>5 TAPEVOL</p> <p>6 TERMINAL</p> <p>7 Reserved for IBM's use</p> <p>Byte 1</p> <p>0 Generic resource name specified.</p> <p>1–7 Reserved for IBM's use</p>
		8	EBCDIC	User ID or group name (OWNER keyword)

Event code dec(hex)	Command	Data length	Format	Description
		1	Binary	Flags for UACC keyword: Bit Authority specified 0 ALTER 1 CONTROL 2 UPDATE 3 READ 4 EXECUTE 5–6 Reserved for IBM's use 7 NONE
		1	Binary	nn (LEVEL keyword)
20(14) (Cont.)	RALTER (Cont.)	1	Binary	Flags for AUDIT keyword: Bit Option specified 0 ALL 1 SUCCESS 2 FAILURES 3 NONE 4–5 Success qualifier codes: '00' READ '01' UPDATE '10' CONTROL '11' ALTER 6–7 FAILURES qualifier codes: '00' READ '01' UPDATE '10' CONTROL '11' ALTER
		1	Binary	*Flags for GLOBALAUDIT keyword: Same format as flags for AUDIT keyword.

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
		2	Binary	<p>Flags for keywords specified:</p> <p>Bit</p> <p>Keyword specified</p> <p>Byte 0</p> <p>0 NOTIFY</p> <p>*1 NONOTIFY</p> <p>2 TVTOC</p> <p>*3 NOTVTOC</p> <p>4 TIMEZONE</p> <p>*5 NOTIMEZONE</p> <p>6 ADDCATEGORY</p> <p>*7 DELCATEGORY</p> <p>Byte 1</p> <p>0 SECLEVEL</p> <p>*1 NOSECLEVEL</p> <p>2 FROM</p> <p>3 FCLASS</p> <p>4 FVOLUME</p> <p>5 FGENERIC</p> <p>6 SECLABEL</p> <p>7 NOSECLABEL</p>
		2	Binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		8	EBCDIC	User ID to be notified when profile denies access
		44	EBCDIC	FROM resource name
		6	EBCDIC	FROM volume volser

Event code dec(hex)	Command	Data length	Format	Description
20(14) (Cont.)	RALTER (Cont.)	8	EBCDIC	FROM class name
		1	Binary	LOGON days:
				Bit
				Day specified
				0
				Sunday
				1
				Monday
				2
				Tuesday
				3
				Wednesday
				4
				Thursday
				5
				Friday
				6
				Saturday
				7
				No keyword
		3	packed	Logon time, packed. If no subkeyword, then binary zeros.
		3	packed	Logoff time, packed. If no subkeyword, then binary zeros.
		3	packed	TIMEZONE value:
				Bit
				Bit value specified
				Byte 0–2
				Signed decimal number
		44	EBCDIC	SECLEVEL name
		8	EBCDIC	SECLABEL name

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
21(15)	RDEFINE	* The data for event code 21 is identical to the data for event code 20, with these exceptions.		
		2	Binary	Flags for keywords specified:
				Bit
				Keyword specified
				Byte 0
				0 DATA
				1 OWNER
				2 UACC
				3 LEVEL
				4 AUDIT
				5 GLOBALAUDIT
				6 ADDVOL
				7 DELVOL
				Byte 1
				0 ADDMEM
				1 DELMEM
				2 APPLDATA
				3 SINGLEDSN
				4 NOSINGLEDSN
				5 WARNING
				6 NOWARNING
				7 WHEN
		2	Binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		2	Binary	Flags for class name:
				Bit
				Option specified
				Byte 0
				0–3 Reserved for IBM's use
				4 DASDVOL
				5 TAPEVOL
				6 TERMINAL
				7 Reserved for IBM's use
				Byte 1
				0 Generic resource name specified
				1–7 Reserved for IBM's use
		8	EBCDIC	User ID or group name (OWNER keyword)

Event code dec(hex)	Command	Data length	Format	Description
21(15) (Cont.)	RDEFINE (Cont.)	1	Binary	Flags for UACC keyword:
				Bit
				Authority specified
				0 ALTER
				1 CONTROL
				2 UPDATE
				3 READ
				4 EXECUTE
				5–6 Reserved for IBM's use
				7 NONE
		1	Binary	nn (LEVEL keyword)
		1	Binary	Flags for AUDIT keyword:
				Bit
				Authority specified
				0 ALL
				1 SUCCESS
				'00' READ
				'01' UPDATE
				'10' CONTROL
				'11' ALTER
				2 FAILURES
				'00' READ
				'01' UPDATE
				'10' CONTROL
				'11' ALTER
				3 NONE
				4–5 SUCCESS qualifier codes
				6–7 FAILURES qualifier codes
		1	Binary	*Reserved for IBM's use

Data type 6

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

Event code dec(hex)	Command	Data length	Format	Description
21(15) (Cont.)	RDEFINE (Cont.)	6	EBCDIC	FROM volume volser
		8	EBCDIC	FROM class name
		1	Binary	LOGON days:
				Bit
				Day specified
				0
				Sunday
				1
				Monday
				2
				Tuesday
				3
				Wednesday
				4
				Thursday
				5
				Friday
				6
				Saturday
				7
				No keyword
		3	packed	Logon time, packed. If no subkeyword, then binary zeros.
		3	packed	Logoff time, packed. If no subkeyword, then binary zeros.
		3	packed	TIMEZONE value:
				Bit
				Option specified
				Byte 0
				0–7
				Reserved for IBM's use
				Byte 1
				0–7
				Reserved for IBM's use
				Byte 2
				0–3
				Reserved for IBM's use
				4–7
				Time zone
		44	EBCDIC	SECLEVEL name
		8	EBCDIC	SECLABEL name
22(16)	RDELETE	2	Binary	Flags for class name:
				Bit
				Option specified
				Byte 0
				0–3
				Reserved for IBM's use
				4
				DASDVOL
				5
				TAPEVOL
				6
				TERMINAL
				7
				Reserved for IBM's use
				Byte 1
				0
				Generic resource name specified
				1–7
				Reserved for IBM's use

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
23(17)	REMOVE	1	Binary	Flags for keywords specified: Bit Keyword specified 0 GROUP 1 OWNER 2–7 Reserved for IBM's use
		1	Binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		8	EBCDIC	User ID (to be removed)
		8	EBCDIC	Group name (GROUP keyword)
		8	EBCDIC	User ID or group name (OWNER keyword)

Event code dec(hex)	Command	Data length	Format	Description
24(18)	SETROPTS	3	Binary	<p>Flags for keywords specified:</p> <p>Bit</p> <p>Option specified</p> <p>Byte 0</p> <p>0 TAPE</p> <p>1 NOTAPE</p> <p>2 INITSTATS</p> <p>3 NOINITSTATS</p> <p>4 SAUDIT</p> <p>5 NOSAUDIT</p> <p>6 STATISTICS</p> <p>7 NOSTATISTICS</p> <p>Byte 1</p> <p>0 AUDIT</p> <p>1 NOAUDIT</p> <p>2 TERMINAL</p> <p>3 NOTERMINAL</p> <p>4 INTERVAL (PASSWORD)</p> <p>5 CMDVIOL</p> <p>6 NOCMDVIOL</p> <p>7 DASD</p> <p>Byte 2</p> <p>0 NODASD</p> <p>1 CLASSACT</p> <p>2 NOCLASSACT</p> <p>3 HISTORY or NOHISTORY</p> <p>4 WARNING or NOWARNING</p> <p>5 REVOKE or NOREVOKE</p> <p>6 NORULES or RULEn</p> <p>7 INACTIVE INTERVAL</p>
		3	Binary	<p>Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.</p>

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
		1	Binary	Flags for STATISTICS or NOSTATISTICS keyword: Bit Option specified Byte 0 0–2 Reserved for IBM's use 3 DATASET 4 DASDVOL 5 TAPEVOL 6 TERMINAL 7 Reserved for IBM's use
		1	Binary	Flags for keywords ignored: Bit Keyword specified 0 MODEL-GDG 1 MODEL-NOGDG 2 MODEL-USER 3 MODEL-NOUSER 4 MODEL-GROUP 5 MODEL-NOGROUP 6 GRPLIST 7 NOGRPLIST
24(18) (Cont.)	SETROPTS (Cont.)	1	Binary	Flags for AUDIT or NOAUDIT keyword: Bit Option specified 0 Reserved for IBM's use 1 GROUP 2 USER 3 DATASET 4 DASDVOL 5 TAPEVOL 6 TERMINAL 7 Reserved for IBM's use

Event code dec(hex)	Command	Data length	Format	Description
		1	Binary	Flags for keywords specified: Bit Option specified 0 MODEL-GDG 1 MODEL-NOGDG 2 MODEL-USER 3 MODEL-NOUSER 4 MODEL-GROUP 5 MODEL-NOGROUP 6 GRPLIST 7 NOGRPLIST
		1	Binary	Change-interval (INTERVAL keyword)
		1	Binary	Flags for TERMINAL keyword: Bit Option specified 0–2 Reserved for IBM's use 3 READ 4–6 Reserved for IBM's use 7 NONE
		1	Binary	Flags for current statistics options after SETROPTS has executed: Bit Option specified 0 Reserved for IBM's use 1 Bypass RACINIT statistics 2 Bypass data set statistics 3 Bypass tape volume statistics 4 Bypass DASD volume statistics 5 Bypass terminal statistics 6 Bypass ADSP attribute 7 EGN in effect

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
		1	Binary	Flags for current audit options after SETROPTS has executed: Bit Option specified 0 Reserved for IBM's use 1 Log group class 2 Log user class 3 Log data set class 4 Log DASD volume class 5 Log tape volume class 6 Log terminal class 7 Reserved for IBM's use
		1	Binary	Reserved for IBM's use
24(18) (Cont.)	SETROPTS (Cont.)	2	Binary	Flags for miscellaneous options after SETROPTS has executed: Bit Option specified Byte 0 0 Perform terminal authorization checking 1 Terminal UACC=NONE (if this bit is off, terminal UACC=READ) 2 Log RACF command violations 3 Log SPECIAL user activity 5–7 Reserved for IBM's use Byte 1 0 Tape volume protection is in effect 1 DASD volume protection is in effect 2 Generic profile processing is in effect for the DATASET class 3 Generic command (GENCMD) processing is in effect for the DATASET class 4 REALDSN is in effect 5 JES-XBMALLRACF is in effect 6 JES-EARLYVERIFY is in effect 7 JES-BATCHALLRACF is in effect
		1	Binary	Maximum password interval
		1	Binary	Password history generation value
		1	Binary	Password revoke value
		1	Binary	Password warning level

Event code dec(hex)	Command	Data length	Format	Description
		80	Binary EBCDIC	<p>Password syntax rules (eight rules). Each rule has the following basic format:</p> <p>Byte</p> <p>Description</p> <p>0 Starting length value</p> <p>1 Ending length value</p> <p>2–9 Character content rules for each of the eight possible positions. The character values are:</p> <p>L = Alphanumeric A = Alphabetic N = Numeric V = Vowel C = Consonant W = No vowels c = Mixed consonant m = Mixed numeric v = Mixed vowel \$ = National s = Special x = Mixed all * = Anything</p>
		1	Binary	User ID inactive interval

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
24(18) (Cont.)	SETROPTS (Cont.)	3	Binary	Flags for keywords specified: Bit Option specified Byte 0 0 ADSP 1 NOADSP 2 GENERIC 3 NOGENERIC 4 GENCMD 5 NOGENCMD 6 GLOBAL 7 NOGLOBAL Byte 1 0 PREFIX 1 NOPREFIX 2 REALDSN 3 NOREALDSN 4 JES-XBMALLRACF 5 JES-NOXBMALLRACF 6 JES-BATCHALLRACF 7 JES-NOBATCHALLRACF Byte 2 0 JES-EARLYVERIFY 1 JES-NOEARLYVERIFY 2 REFRESH 3 PROTECTALL-WARNING 4 PROTECTALL-FAILURE 5 NOPROTECTALL 6 EGN in effect 7 NOEGN in effect
		3	Binary	Flags for keywords specified but ignored because of insufficient authority: Same format as flags for keywords specified.
		8	EBCDIC	Single-level data set name prefix

Event code dec(hex)	Command	Data length	Format	Description
		3	Binary	Flags for keywords specified: Bit Keyword specified Byte 0 0 TAPEDSN 1 NOTAPEDSN 2 NOEOS 3 EOS 4 EOS-SECLEVEL 5 EOS-NOSECLEVEL 6 RETPD 7 WHEN Byte 1 0 NOWHEN 1 OPERAUDIT 2 NOOPERAUDIT 3 RVAR SWITCH 4 RVAR ACTIVE/INACTIVE 5 ERASE-ALL 6 APPLAUDIT 7 NOAPPLAUDIT Byte 2 0-7 Reserved for IBM's use
24(18) (Cont.)	SETROPTS (Cont.)	3	Binary	Flags for keywords specified but ignored because of insufficient authority: Same format as flags for keywords specified.
		1	Binary	Erase on scratch security level
		2	Binary	Retention period

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
		1	Binary	Flags for miscellaneous options after SETROPTS processing: Bit Option specified Byte 0 0 PROTECTALL-WARNING 1 PROTECTALL-FAILURES 2 EOS 3 EOS-SECLEVEL 4 TAPEDSN 5 WHEN 6 EOS ALL IN EFFECT (erase everything) 7 Reserved for IBM's use

Event code dec(hex)	Command	Data length	Format	Description
24(18) (Cont.)	SETROPTS (Cont.)	5	Binary	<p>Flags for keywords specified:</p> <p>Bit</p> <p>Option specified</p> <p>Byte 0</p> <p>0–7</p> <p>Reserved for IBM's use</p> <p>Byte 1</p> <p>0</p> <p>GENLIST</p> <p>1</p> <p>NOGENLIST</p> <p>2</p> <p>RACLIST</p> <p>3</p> <p>NORACLIST</p> <p>4</p> <p>SECLEVELAUDIT</p> <p>5</p> <p>NOSECLEVELAUDIT</p> <p>6</p> <p>SECLABELAUDIT</p> <p>7</p> <p>NOSECLABELAUDIT</p> <p>8</p> <p>SECLABELCONTROL</p> <p>9</p> <p>NOSECLABELCONTROL</p> <p>10</p> <p>MLQUIET</p> <p>11</p> <p>NOMLQUIET</p> <p>12</p> <p>MLSTABLE</p> <p>13</p> <p>NOMLSTABLE</p> <p>14</p> <p>GENERICOWNER</p> <p>15</p> <p>NOGENERICOWNER</p> <p>16</p> <p>SESSIONINTERVAL</p> <p>17</p> <p>NOSESSIONINTERVAL</p> <p>18</p> <p>JES NJEUSERID (user ID)</p> <p>19</p> <p>JES UNDEFINEDUSER (user ID)</p> <p>20</p> <p>COMPATMODE</p>

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
24 (18) (Cont.)	SETROPTS (Cont.)	5	Binary	21 NOCOMPATMODE
				22 MLS WARNING
				23 MLS FAILURES
				24 NOMLS
				25 MLACTIVE WARNING
				26 MLACTIVE FAILURES
				27 NOMLACTIVE
				28 CATDSNS WARNING
				29 CATDSNS FAILURES
				30 NOCATDSNS
				31 LOGOPTIONS
		4	Binary	Flags for keywords specified but ignored because of insufficient authority: Same format as flags for keywords specified.
		1	Binary	SECLEVEL audit value (auditing occurs for all resources having at least this value)
		2	Binary	SESSIONINTERVAL interval
		1	Binary	Log options for data set
				Bit
				Keyword specified
				0 ALWAYS
				1 NEVER
				2 SUCCESSES
				3 FAILURES
				4 DEFAULT
				5–7 Reserved for IBM's use

Event code dec(hex)	Command	Data length	Format	Description
		2	Binary	Current SETROPTS options for multilevel security
				Bit
				Keyword specified
				0 SECLABELAUDIT
				1 SECLABELCONTROL
				2 MLQUIET
				3 MLSTABLE
				4 GENERICOWNER
				5 COMPATMODE
				6 MLS WARNING
				7 MLS FAILURES
				8 MLACTIVE WARNING
				9 MLACTIVE FAILURES
				10 CATDSNS WARNING
				11 CATDSNS FAILURES
				12 APPLAUDIT
				13 ADDCREATOR
				14 ENHANCEDGENERICOWNER
				15 Reserved for IBM's use
		8	EBCDIC	User ID for JES NJEUSERID
		8	EBCDIC	User ID for JES UNDEFINEDUSER
		1	Binary	Password MINCHANGE interval value
		1	EBCDIC	Reserved for IBM's use

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
		4	Binary	Flags for keywords specified Bit Keyword specified 0 Primary language specified 1 Secondary language specified 2 ADDCREATOR specified 3 NOADDCREATOR specified 4 LIST specified 5 KERBLVL specified 6 ENHANCEDGENERICOWNER specified 7 Reserved for IBM's use 8 Password MINCHANGE specified 9 Password MIXEDCASE specified 10 Password NOMIXEDCASE specified 11 Password SPECIALCHARS specified 12 Password NOSPECIALCHARS specified 13 Password ALGORITHM specified 14 Password NOALGORITHM specified 15 Password PHRASEINT specified 16 MLFSOBJ(ACTIVE) specified 17 MLFSOBJ(INACTIVE) specified 18 MLIPCOBJ(ACTIVE) specified 19 MLFSOBJ(INACTIVE) specified 20 MLNAMES specified
24(18) (Cont.)	SETROPTS (Cont.)			21 NOMLNAMES specified 22 SECLBYSYSTEM specified 23 NOSECLBYSYSTEM specified 24 KDFAES specified 25–31 Reserved for IBM's use
		4	Binary	Flags for keywords specified but ignored because of insufficient authority: same format as flags for keywords specified.
		3	EBCDIC	Primary language default
		3	EBCDIC	Secondary language default

Event code dec(hex)	Command	Data length	Format	Description
		1	Binary	Flags for asterisk (*) specified Bit Keyword specified 0 Asterisk (*) specified for GENERIC 1 Asterisk (*) specified for GLOBAL 2 Asterisk (*) specified for AUDIT 3 Asterisk (*) specified for STATISTICS 4 Asterisk (*) specified for CLASSACT 5 Asterisk (*) specified for GENCMD 6 Asterisk (*) specified for LOGOPTIONS DEFAULT 7 Reserved for IBM's use
		1	Binary	KERBLVL setting
		1	Binary	Current multilevel security options Bit Keyword specified 0 MLFSOBJ is active 1 MLIPCOBJ is active 2 MLNAMES is active 3 SECLBYSYSTEM is active 4–7 Reserved for IBM's use
		1	Binary	Current minimum password change interval (MINCHANGE)
		1	Binary	Current options Bit Option 0 Mixed case passwords are allowed 1 Special characters are allowed in passwords 2–7 Reserved for IBM's use
		1	Binary	Password algorithm in effect Bit Meaning 0 Existing algorithm as indicated by ICHDEX01 (masking, DES, or installation-defined) 1 KDFAES
		2	Binary	Password Phrase change-interval (PHRASEINT) keyword)
		73	EBCDIC	Reserved for IBM's use

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
25(19)	RVARY	1	Binary	Flags for keywords specified: Bit Keyword specified 0 ACTIVE 1 INACTIVE 2 NOTAPE 3 NOCLASSACT 4 SWITCH 5 DATASET 6 LIST 7 NOLIST
		1	Binary	Flags for other violations: Bit Violation 0 Command denied by operator 1 Nonzero code returned from RACF manager during ACTIVE processing 2–7 Reserved for IBM's use
		1	Binary	Flags for other keywords specified: Bit Keyword specified 0 DATASHARE 1 NODATASHARE
59(3B)	RACLINK	20	EBCDIC	Phase identifier (1 of 3 values: LOCAL ISSUANCE, TARGET PROCESSING, or TARGET RESPONSE)

Event code dec(hex)	Command	Data length	Format	Description
		2	Binary	Flags for keywords specified: Bit Option specified Byte 0 0 DEFINE 1 UNDEFINE 2 APPROVE 3–7 Reserved for IBM's use Byte 1 0 PEER 1 MANAGED 2 PWSYNC 3 NOPWSYNC 4 Password supplied 5–7 Reserved for IBM's use
		2	Binary	Reserved for IBM's use
		8	EBCDIC	Issuing node
		8	EBCDIC	Issuing user ID
		8	EBCDIC	Source user ID for association (from ID keyword)
		8	EBCDIC	Target node name
		8	EBCDIC	Target user ID
		8	EBCDIC	Target authorization ID (ID under whose authority the association was established)
		4	EBCDIC	Originating system's SMF ID from where LOCAL ISSUANCE occurred
		4	Binary	Original time stamp (local time) from when LOCAL ISSUANCE occurred
		4	Packed	Original date when LOCAL ISSUANCE occurred Note: The preceding 3 fields contain the LOCAL ISSUANCE information for all 3 phases.

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
		1	Binary	<div>Status flags:</div> <div><div>Bit</div><div>Status</div></div> <div>Byte 0</div> <div><div>0</div><div>Association established</div></div> <div><div>1</div><div>Association pending</div></div> <div><div>2</div><div>Association deleted</div></div> <div><div>3</div><div>Password supplied is not valid</div></div> <div><div>4</div><div>Valid password supplied</div></div> <div><div>5</div><div>Expired password supplied</div></div> <div><div>6</div><div>Revoked user ID</div></div> <div><div>7</div><div>Reserved for IBM's use</div></div> <div><div>Note:</div><div>When the event code qualifier is 0, and the status flags indicate that no password was supplied and that the association is established, an authorization user ID was used from the association list. If the status flags indicate that no password was supplied and the association is pending, no user ID in the authorization list had the appropriate authority or no association list exists.</div></div>

Event code dec(hex)	Command	Data length	Format	Description
66(42)	RACDCERT	4	Binary	<p>Flags for keywords specified:</p> <p>Bit</p> <p>Keyword specified</p> <p>Byte 0</p> <p>0 ADD</p> <p>1 ALTER</p> <p>2 DELETE</p> <p>3 CONNECT</p> <p>4 REMOVE</p> <p>5 SITE</p> <p>6 CERTAUTH</p> <p>7 ICSF</p> <p>Byte 1</p> <p>0 TRUST</p> <p>1 NOTRUST</p> <p>2 ADDRING</p> <p>3 DELRING</p> <p>4 USAGE(PERSONAL)</p> <p>5 USAGE(SITE)</p> <p>6 USAGE(CERTAUTH)</p> <p>7 DEFAULT</p> <p>Byte 2</p> <p>0 CONNECT(SITE)</p> <p>1 CONNECT(CERTAUTH)</p> <p>2 GENCERT</p> <p>3 EXPORT</p> <p>4 GENREQ</p> <p>5 SIGNWITH(CERTAUTH... specified)</p> <p>6 SIGNWITH(SITE... specified)</p> <p>7 PASSWORD</p>

Data type 6

Event code dec(hex)	Command	Data length	Format	Description
66(42) (Cont.)	RACDCERT (Cont.)	4	Binary	Byte 3
				0 MAP
				1 ALTMAP
				2 DELMAP
				3 MULTIID
				4 HIGHTRUST
				5 PCICC
				6 DSA
				7 FROMICSF
		8	EBCDIC	User ID (from ID keyword on RACDCERT)
		44	EBCDIC	Data set name
		32	EBCDIC	Label name
		8	EBCDIC	User ID (from ID sub-keyword)
		32	EBCDIC	WITHLABEL
		4	Binary	SIZE
		10	EBCDIC	NOTBEFORE(date) in the format yyyy/mm/dd
		8	EBCDIC	NOTBEFORE(time) in the format hh:mm:ss
		10	EBCDIC	NOTAFTER(date) in the format yyyy/mm/dd
		8	EBCDIC	NOTAFTER(time) in the format hh:mm:ss
		1	Binary	FORMAT
				X'01' CERTB64
				X'02' CERTDER
				X'03' PKCS12B64
				X'04' PKCS12DER
				X'05' PKCS7B64
				X'06' PKCS7DER

Event code dec(hex)	Command	Data length	Format	Description
66(42) (Cont.)	RACDCERT (Cont.)	4	Binary	<p>More flags for keywords specified:</p> <p>Bit</p> <p>Keyword specified</p> <p>Byte 0</p> <p>0 ALTIP</p> <p>1 ALTEMAIL</p> <p>2 ALTDOMAIN</p> <p>3 ALTURI</p> <p>4 KUHANDSHAKE</p> <p>5 KUDATAENCR</p> <p>6 KUDOCSIGN</p> <p>7 KUCERTSIGN</p> <p>Byte 1</p> <p>0 REKEY</p> <p>1 ROLLOVER</p> <p>2 FORCE</p> <p>3 ADDTOKEN</p> <p>4 DELTOKEN</p> <p>5 BIND</p> <p>6 UNBIND</p> <p>7 IMPORT</p> <p>Byte 2</p> <p>0 NISTECC</p> <p>1 BPECC</p> <p>2 KUKEYAGREE</p> <p>3 RSA</p> <p>4 PKDS</p> <p>5 TOKEN</p> <p>6 SIGATTRRSS</p> <p>7 PBEAES</p> <p>Byte 3</p> <p>0–7 Reserved for IBM's use</p>
		4	Binary	SEQNUM

SMF record type 81

Event code dec(hex)	Command	Data length	Format	Description
87(57)	RACMAP	4	Binary	Flags for keywords specified: Bit Keyword specified Byte 0 0 MAP 1 DELMAP 2 QUERY 3–7 Reserved for IBM's use Byte 1 0–7 Reserved for IBM's use Byte 2 0–7 Reserved for IBM's use Byte 3 0–7 Reserved for IBM's use
		8	EBCDIC	User ID
		32	EBCDIC	Label name

Record type 81: RACF initialization record

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

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

The format of record type 81 is:

Table 4. Initialization record (type 81)					
Offset (Dec.)	Offset (Hex.)	Name	Length	Format	Description
0	0	SMF81LEN	2	Binary	Record length.
2	2	SMF81SEG	2	Binary	Segment descriptor.

Table 4. Initialization record (type 81) (continued)

Offset (Dec.)	Offset (Hex.)	Name	Length	Format	Description
4	4	SMF81FLG	1	Binary	System indicator Bit Meaning when set 0-2 Reserved for IBM's use 3 MVS/ 4 MVS/ 5 MVS/ 6 VS2 7 Reserved for IBM's use Note: For MVS/, bits 3, 4, 5, and 6 are on.
5	5	SMF81RTY	1	Binary	Record type: 81 (X'51').
6	6	SMF81TME	4	Binary	Time of day, in hundredths of a second, that the record was moved to the SMF buffer.
10	A	SMF81DTE	4	packed	Date that the record was moved to the SMF buffer, in the form <i>OcyyddF</i> (where <i>F</i> is the sign).
14	E	SMF81SID	4	EBCDIC	System identification (from the SID parameter).
18	12	SMF81RDS	44	EBCDIC	Data set name of the RACF database for this IPL (blanks if RACF is not active).
62	3E	SMF81RVL	6	EBCDIC	Volume identification of RACF database. If the database is split among several DASD volumes, this field equals the first primary data set. If RACF is not active, this field is blank.
68	44	SMF81RUN	3	EBCDIC	Unit name of RACF database; blanks if RACF is not active. Note: If the master RACF primary database is on a device whose address is greater than X'FFF', the field contains 'UCB' instead of the EBCDIC device name.
71	47	SMF81UDS	44	EBCDIC	Data set name of the user attribute data set (UADS) data set for this IPL.
115	73	SMF81UVL	6	EBCDIC	Volume identification of the user attribute data set (UADS) data set.

SMF record type 81

Table 4. Initialization record (type 81) (continued)

Offset (Dec.)	Offset (Hex.)	Name	Length	Format	Description
121	79	SMF81OPT	1	Binary	Options indicator Bit Meaning when set 0 No RACROUTE REQUEST=VERIFY statistics are recorded 1 No DATASET statistics are recorded 2 RACROUTE REQUEST=VERIFY preprocessing exit routine, ICHRIX01, is active 3 RACROUTE REQUEST=AUTH preprocessing exit routine, ICHRCX01, is active 4 RACROUTE REQUEST=DEFINE preprocessing exit routine, ICHRDY01, is active 5 RACROUTE REQUEST=VERIFY postprocessing exit routine, ICHRIX02, is active 6 RACROUTE REQUEST=AUTH postprocessing exit routine, ICHRCX02, is active 7 New-password exit routine, ICHPW01, is active
122	7A	SMF81OP2	1	Binary	Options indicator 2 Bit Meaning when set 0 No tape volume statistics are recorded 1 No DASD volume statistics are recorded 2 No terminal statistics are recorded 3 Command exit routine ICHCNX00 is active 4 Command exit routine ICHCCX00 is active 5 ADSP is not active 6 Encryption exit routine, ICHDEX01 is active 7 Naming convention table, ICHNCV00 is present.

Table 4. Initialization record (type 81) (continued)

Offset (Dec.)	Offset (Hex.)	Name	Length	Format	Description
123	7B	SMF81OP3	1	Binary	Options indicator 3 Bit Meaning when set 0 Tape volume protection is in effect. 1 No duplicate data set names are to be defined 2 DASD volume protection is in effect 3 Record contains version indicator 4 RACROUTE REQUEST=FASTAUTH preprocessing exit routine, ICHRFX01, is active 5 RACROUTE REQUEST=LIST pre- and postprocessing exit routine, ICHRLX01, is active 6 RACROUTE REQUEST=LIST selection exit routine, ICHRLX02, is active 7 RACROUTE REQUEST=DEFINE postprocessing exit routine, ICHRDY02, is active.
124	7C	SMF81AOP	1	Binary	Audit options Bit Meaning when set 0 User class profile changes are being logged 1 Group class profile changes are being logged 2 Data set class profile changes are being logged 3 Tape volume class profile changes are being logged 4 DASD volume class profile changes are being logged 5 Terminal class profile changes are being logged 6 RACF command violations are being logged 7 SPECIAL user activity is being logged.
125	7D	SMF81AO2	1	Binary	Audit options 2 Bit Meaning when set 0 Operation user activity 1 Audit by security level is in effect 2-7 Reserved for IBM's use

Table 4. Initialization record (type 81) (continued)

Offset (Dec.)	Offset (Hex.)	Name	Length	Format	Description
126	7E	SMF81TMO	1	Binary	<p>Terminal verification options indicator</p> <p>Bit</p> <p>Meaning when set</p> <p>0</p> <p>Terminal authorization checking is in effect</p> <p>1</p> <p>Universal access for undefined terminals is NONE; if not set, UACC=READ</p> <p>2</p> <p>REALDSN is in effect</p> <p>3</p> <p>JES-XBMALLRACF is in effect</p> <p>4</p> <p>JES-EARLYVERIFY is in effect</p> <p>Note: Early verification is always done even if this bit indicates that JES-EARLYVERIFY is not in effect. For more information, see <i>JES user ID early verification</i> in <i>z/OS Security Server RACF Security Administrator's Guide</i>.</p> <p>5</p> <p>JES-BATCHALLRACF is in effect</p> <p>6</p> <p>RACROUTE REQUEST=FASTAUTH postprocessing exit routine, ICHRF02, is active</p> <p>7</p> <p>Reserved for IBM's use</p>
127	7F	SMF81PIV	1	Binary	Maximum password interval (0-254).
128	80	SMF81REL	2	Binary	Offset to the first relocate section from the beginning of the record header.
130	82	SMF81CNT	2	Binary	Number of relocate sections.
132	84	SMF81VER	1	Binary	Version indicator (6 = RACF Version 1, Release 7). As of RACF 1.8.1, SMF81VRM is used instead.
133	85	SMF81QL	8	EBCDIC	Single-level data set name.
141	8D	SMF81OP4	1	Binary	<p>Options indicator 4</p> <p>Bit</p> <p>Meaning when set</p> <p>0</p> <p>TAPEDSN is in effect</p> <p>1</p> <p>PROTECT-ALL is in effect</p> <p>2</p> <p>PROTECT-ALL warning is in effect</p> <p>3</p> <p>ERASE-ON-SCRATCH is in effect</p> <p>4</p> <p>ERASE-ON-SCRATCH by SECLEVEL is in effect</p> <p>5</p> <p>ERASE-ON-SCRATCH for all data sets is in effect</p> <p>6</p> <p>Enhanced generic naming is in effect</p> <p>7</p> <p>Record contains a version, release, and modification number (see SMF81VRM).</p>

Table 4. Initialization record (type 81) (continued)

Offset (Dec.)	Offset (Hex.)	Name	Length	Format	Description
142	8E	SMF81OP5	1	Binary	Options indicator 5 Bit Meaning when set 0 Access control by program is in effect 1 ACEE compression/expansion exit IRRACX01 is active 2 RACROUTE REQUEST=FASTAUTH postprocessing exit ICHRFX04 is active 3 RACROUTE REQUEST=FASTAUTH preprocessing exit ICHRFX03 is active 4 SETROPTS NOADDCREATOR is active 5 IRREVSX01 exit is active Note: The IRREVSX01 exit point is defined to dynamic exit services. Bit 5 of SMF81OP5 indicates that an exit routine was active for this exit point at the time of the last IPL when the SMF record was written. The status can change either way multiple times throughout the life of the IPL. See the SET PROG operator command in <i>z/OS MVS System Commands</i> and the CSVDYNEX macro in <i>z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN</i> for more information. 6 ACEE compression/expansion exit IRRACX02 is active 7 Password exit routine, ICHDEX11 is active
143	8F	SMF81RPD	2	Binary	System retention period in effect.
145	91	SMF81SLV	1	Binary	Security level for ERASE-ON-SCRATCH in effect.
146	92	SMF81SLC	1	Binary	Security level for auditing in effect.

Table 4. Initialization record (type 81) (continued)

Offset (Dec.)	Offset (Hex.)	Name	Length	Format	Description
147	93	SMF81VRM	4	EBCDIC	FMID for RACF
					2020 RACF 2.2 and OS/390 Security Server (RACF) V1 R2
					2030 OS/390 Security Server (RACF) V1 R3
					2040 OS/390 Security Server (RACF) V2 R4
					2060 OS/390 Security Server (RACF) V2 R6
					2608 OS/390 Security Server (RACF) V2 R8
					7703 OS/390 Security Server (RACF) V2 R10 and z/OS Security Server (RACF) V1 R1
					7705 z/OS Security Server (RACF) V1 R2
					7706 z/OS Security Server (RACF) V1 R3
					7707 z/OS Security Server (RACF) V1 R4
					7708 z/OS Security Server (RACF) V1 R5
					7709 z/OS Security Server (RACF) V1 R6
					7720 z/OS Security Server (RACF) V1 R7
					7730 z/OS Security Server (RACF) V1 R8
					7740 z/OS Security Server (RACF) V1 R9
					7750 z/OS Security Server (RACF) V1 R10
					7760 z/OS Security Server (RACF) V1 R11
					7770 z/OS Security Server (RACF) V1 R12
					7780 z/OS Security Server (RACF) V1 R13
					7790 z/OS Security Server (RACF) V2 R1
					77A0 z/OS Security Server (RACF) V2 R2
					77B0 z/OS Security Server (RACF) V2 R3
					77C0 z/OS Security Server (RACF) V2 R4
					77D0 z/OS Security Server (RACF) V2 R5
					77E0 z/OS Security Server (RACF) V3 R1
					77F0 z/OS Security Server (RACF) V3 R2

Table 4. Initialization record (type 81) (continued)

Offset (Dec.)	Offset (Hex.)	Name	Length	Format	Description
151	97	SMF81BOP	1	Binary	SETROPTS options. Bit Meaning when set 0 SECLABELCONTROL is in effect 1 CATDSNS is in effect 2 MLQUIET is in effect 3 MLSTABLE is in effect 4 MLS is in effect 5 MLACTIVE is in effect 6 GENERICOWNER is in effect 7 SECLABELAUDIT is in effect.
152	98	SMF81SIN	2	Binary	Partner LU-verification session key interval.
154	9A	SMF81JSY	8	EBCDIC	JES NJE NAME user ID.
162	A2	SMF81JUN	8	EBCDIC	JES UNDEFINEDUSER user ID.
170	AA	SMF81BOX	1	Binary	SETROPTS option extensions. Bit Meaning when set 0 COMPATMODE is in effect 1 CATDSNS failures are in effect 2 MLS failures are in effect 3 MLACTIVE failures are in effect 4 APPLAUDIT in effect 5 Zero (0) equals default RVAR SWITCH password in effect. One (1) equals installation-defined RVAR SWITCH password in effect. 6 Zero (0) equals default RVAR STATUS password in effect. One (1) equals installation-defined RVAR STATUS password in effect. 7 ENHANCEDGENERICOWNER in effect
171	AB	SMF81PRI	3	EBCDIC	Default primary language for an installation.
172	AC	SMF81SEC	3	EBCDIC	Default secondary language for an installation.
177	B1	SMF81KBL	1	Binary	Level of KERB segment processing in effect.
178	B2	SMF81PMN	1	Signed	Minimum days between password changes

Table 4. Initialization record (type 81) (continued)

Offset (Dec.)	Offset (Hex.)	Name	Length	Format	Description
179	B3	SMF81OP6	1	Binary	Options indicator 6 Bit Meaning when set 0 Mixed case passwords 1 New password phrase installation exit is active 2 Field validation exit point (IRRVAF01) for custom fields is active Note: The IRRVAF01 exit point is defined to dynamic exit services. Bit 2 of SMF81OP6 indicates that an exit routine was active for this exit point at the time of the last IPL when the SMF record was written. The status can change either way multiple times throughout the life of the IPL. See z/OS Security Server RACF System Programmer's Guide for more information. 3 Special characters allowed in passwords 4–7 Reserved for IBM's use
180	B4	SMF81ML2	1	Binary	More SETROPTS options Bit Meaning when set 0 MLFSOBJ is active 1 MLIPCOBJ is active 2 MLNAMES is active 3 SECLBYSYSTEM is active 4–7 Reserved for IBM's use
181	B5	SMF81ALG	1	Binary	Password encryption algorithm in effect. Value Meaning 0 Indicates LEGACY 1 Indicates KDFAES
182	B6	SMF81VXC	8	EBCDIC	VMXEVENT control profile is in effect
190	BE	SMF81VXA	8	EBCDIC	VMXEVENT audit profile is in effect
198	C6	SMF81PHI	2	Binary	Password phrase interval
200	C8		55	Reserved.	
Relocate section:					
0	0	SMF81DTP	1	Binary	Data type.
1	1	SMF81DLN	1	Binary	Length of data that follows.
2	2	SMF81DTA	1-255	mixed	Data.

Record type 83: Security events

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

Notes:

1. Subtype 1 - Record type 83 subtype 1 is a RACF processing record for auditing data sets that are affected by a RACF command (ADDSD, ALTDSD, and DELDSD) that caused the security label to be changed. These records are generated when SETROPTS MACTIVE is in effect and a RACF command (ALTDSD, ADDSD, DELDSD) has been issued that changed the security label of a data set profile. The SMF type 83 subtype 1 record contains the names of the cataloged data sets affected by the security label change.

A link value is contained in both the SMF type 80 record for the RACF command and the SMF type 83 subtype 1 record. The link value is used to connect the list of data set names that are affected by the security label change with the RACF command that caused the change.

The event codes and qualifiers for record type 83 subtype 1 are the same as for type 80 records.

2. Subtype 2 - SMF type 83 subtype 2 records contain Enterprise Identity Mapping (EIM) audit data.
3. Subtype 3 - SMF type 83 subtype 3 records contain LDAP audit data.
4. Subtype 4 - SMF type 83 subtype 4 records contain information from the R_auditx remote auditing service. For more information about SMF type 83 audit records, see [R_auditx \(IRRSAX00 or IRRSAX64\): Audit a security-related event in z/OS Security Server RACF Callable Services](#).
5. Subtype 5 - SMF type 83 subtype 5 records contain WebSphere® audit data.
6. Subtype 6 - SMF type 83 subtype 6 records contain TKLM audit data.
7. Subtype 7 - SMF type 83 subtype 7 records contain IBM Z Multi-Factor Authentication data. For more information about record type 83 subtype 7, see [IBM MFA SMF Record type 83 subtype 7 records \(www.ibm.com/docs/en/SSNR6Z_2.2.0/azfi100/azf_smf_ref.htm\)](#) in *IBM Z Multi-Factor Authentication Installation and Customization*.
8. Subtype 8 - SMF type 83 subtype 8 records contain details about data set access anomalies. For more information, see the description of the Subtype 8 Anomaly relocate section in Appendix A of *IBM Threat Detection for z/OS Guide*.

The format is:

Offset (dec)	Offset (hex)	Name	Length	Format	Description
0	0	SMF83LEN	2	Binary	Record length.
2	2	SMF83SEG	2	Binary	Segment descriptor.

SMF record type 83

Offset (dec)	Offset (hex)	Name	Length	Format	Description
4	4	SMF83FLG	1	Binary	<p>System indicator</p> <p>Bit</p> <p>Meaning when set</p> <p>0 Subsystem identification follows system identification</p> <p>1 Subtypes used</p> <p>2 Reserved for IBM's use</p> <p>3 MVS/</p> <p>4 MVS/</p> <p>5 MVS/</p> <p>6 VS2</p> <p>7 Reserved for IBM's use.</p> <p>Note: For MVS/, bits 3, 4, 5, and 6 are on.</p>
5	5	SMF83RTY	1	Binary	Record type: 83 (X'53').
6	6	SMF83TME	4	Binary	Time of day, in hundredths of a second, that the record was moved to the SMF buffer.
10	A	SMF83DTE	4	EBCDIC	Date that the record was moved to the SMF buffer, in the form <i>OcyyddF</i> (where <i>F</i> is the sign).
14	E	SMF83SID	4	EBCDIC	System identification (from the SID parameter).
18	12	SMF83SSI	4	EBCDIC	Subsystem identification RACF.
22	16	SMF83TYP	2	Binary	<p>Record subtype</p> <p>1 See “Subtype 1” on page 147</p> <p>2 See “Subtype 2 and above” on page 152</p>
24	18	SMF83TRP	2	Binary	Number of triplets.
26	1A	SMF83XXX	2		Reserved for IBM's use.
28	1C	SMF83OPD	4	Binary	Offset to product section.
32	20	SMF83LPD	2	Binary	Length of product section.
34	22	SMF83NPD	2	Binary	Number of product sections.
36	24	SMF83OD1	4	Binary	Offset to security section.
40	28	SMF83LD1	2	Binary	Length of security section.
42	2A	SMF83ND1	2	Binary	Number of security sections.
44	2C	SMF83OD2	4	Binary	Offset to relocate section.
48	30	SMF83LD2	2	Binary	Length of relocate section.
50	32	SMF83ND2	2	Binary	Number of relocate sections.
Product section: See “Product section” on page 147 for details.					
Security section: See “Security section” on page 147 for details.					
Relocate sections: See “Relocate sections” on page 154 for details.					

Product section

The product section exists in all SMF type 83 records. It is completed for subtype 1 records.

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

The product section is mapped in the following table.

Table 5. RACF SMF type 83 record product section					
Offsets					
Dec.	Hex.	Name	Length	Format	Description
0	0	SMF83RVN	4	EBCDIC	Product version, release, and modification level number.
4	4	SMF83PNM	4	EBCDIC	Product name

Security section

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

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

- The user identity or identities used by the product or component for purposes of the authentication or authorization request
- The authority required for the request to succeed
- The authority the user has
- The reasons for logging the event
 1. includes the user identity used to determine why to log
 2. includes the resource used to determine why to log

Note: In general, RACF searches for reasons for auditing an event until it finds one, then audits without looking for more reasons that might also have caused auditing. This means that most RACF SMF records will show only one reason for auditing, even though several might apply (and in a few cases, more than one might actually be shown in the record). There are many places in RACF that audit, and the order of checking is not the same in all places, so the audit reason that will be used is not entirely predictable. In some cases it would not even be possible for RACF to look for additional potential audit reasons without causing adverse performance impact to the system. For example, SPECIAL users are often granted access to a resource without even reading the resource profile that protects it, so no information is available about what auditing options the profile might have requested.

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

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

Subtype 1

Offset (Dec.)	Offset (Hex.)	Name	Length	Format	Description
Security section:					
0	0	SMF83LNK	4	Binary	Same LINK value as that for the SMF type 80 record for the associated command. Connects the data set names in type 83 records with the RACF command that caused the security label change.

SMF record type 83

Offset (Dec.)	Offset (Hex.)	Name	Length	Format	Description
4	4	SMF83DES	2	Binary	<p>Descriptor flags</p> <p>Bit</p> <p>Meaning when set</p> <p>0 The event is a violation</p> <p>1 User is not defined to RACF</p> <p>2 Record contains a version indicator (see SMF83VER)</p> <p>3 The event is a warning</p> <p>4 Record contains a version, release, and modification level number (see SMF83VRM)</p> <p>5-15 Reserved for IBM's use.</p>
6	6	SMF83EVT	1	Binary	Event code.
7	7	SMF83EVQ	1	Binary	Event code qualifier.
8	8	SMF83USR	8	EBCDIC	Identifier of the user associated with this event (jobname is used if the user is not defined to RACF).
16	10	SMF83GRP	8	EBCDIC	Group to which the user was connected (stepname is used if the user is not defined to RACF).
24	18	SMF83REL	2	Binary	Offset to the first relocate section from beginning of record header.
26	1A	SMF83CNT	2	Binary	Count of the number of relocate sections.
28	1C	SMF83ATH	1	Binary	<p>Authorities used for executing commands or accessing resources</p> <p>Bit</p> <p>Meaning when set</p> <p>0 Normal authority check (resource access)</p> <p>1 SPECIAL attribute (command processing)</p> <p>2 OPERATIONS attribute (resource access, command processing)</p> <p>3 AUDITOR attribute (command processing)</p> <p>4 Installation exit processing (resource access)</p> <p>5 Failsoft processing (resource access)</p> <p>6 Bypassed-user ID = *BYPASS* (resource access)</p> <p>7 Trusted attribute (resource access).</p>

Offset (Dec.)	Offset (Hex.)	Name	Length	Format	Description
29	1D	SMF83REA	1	Binary	Reason for logging. These flags indicate the reason RACF produced the SMF record Bit Meaning when set 0 SETROPTS AUDIT(class) changes to this class of profile are being audited. 1 User being audited 2 SPECIAL users being audited 3 Access to the resource is being audited because of the AUDIT option (specified when profile created or altered by a RACF command), a logging request from the RACHECK exit routine, or because the operator granted access during failsoft processing. 4 RACINIT failure 5 This command is always audited 6 Violation detected in command and CMDVIOL is in effect 7 Access to entity being audited because of GLOBALAUDIT option.
30	1E	SMF83TLV	1	Binary	Terminal level number of foreground user (zero if not available).
31	1F	SMF83ERR	1	Binary	Command processing error flag Bit Meaning when set 0 Command had error and RACF could not back out some changes 1 No profile updates were made because of error in RACF processing 2-7 Reserved for IBM's use.
32	20	SMF83TRM	8	EBCDIC	Terminal ID of foreground user (zero if not available).
40	28	SMF83JBN	8	EBCDIC	Job name. For RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX records for batch jobs, this field can be zero.
48	30	SMF83RST	4	Binary	Time, in hundredths of a second that the reader recognized the JOB statement for this job for RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX records for batch jobs, this field can be zero.
52	34	SMF83RSD	4	packed	Date the reader recognized the JOB statement for this job in the form 0cyydd <i>F</i> (where <i>F</i> is the sign) for RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX records for batch jobs, this field can be zero.
56	38	SMF83UID	8	EBCDIC	User identification field from the SMF common exit parameter area. For RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX records for batch jobs, this field can be zero.
64	40	SMF83VER	1	Binary	Version indicator 8 = Version 1, Release 8 or later. As of RACF 1.8.1, SMF83VRM is used instead.

SMF record type 83

Offset (Dec.)	Offset (Hex.)	Name	Length	Format	Description
65	41	SMF83RE2	1	Binary	Additional reasons for logging Bit Meaning when set 0 Security level control for auditing 1 Auditing by LOGOPTIONS 2 Audited because of SETROPTS SECLABELAUDIT 3 Class being audited because of SETROPTS COMPATMODE 4-7 Reserved for IBM's use.

Offset (Dec.)	Offset (Hex.)	Name	Length	Format	Description
66	42	SMF83VRM	4	EBCDIC	FMID for RACF 2020 RACF 2.2 and OS/390 Security Server (RACF) V1 R2 2030 OS/390 Security Server (RACF) V1 R3 2040 OS/390 Security Server (RACF) V2 R4 2060 OS/390 Security Server (RACF) V2 R6 2608 OS/390 Security Server (RACF) V2 R8 7703 OS/390 Security Server (RACF) V2 R10 and z/OS Security Server (RACF) V1 R1 7705 z/OS Security Server (RACF) V1 R2 7706 z/OS Security Server (RACF) V1 R3 7707 z/OS Security Server (RACF) V1 R4 7708 z/OS Security Server (RACF) V1 R5 7709 z/OS Security Server (RACF) V1 R6 7720 z/OS Security Server (RACF) V1 R7 7730 z/OS Security Server (RACF) V1 R8 7740 z/OS Security Server (RACF) V1 R9 7750 z/OS Security Server (RACF) V1 R10 7760 z/OS Security Server (RACF) V1 R11 7770 z/OS Security Server (RACF) V1 R12 7780 z/OS Security Server (RACF) V1 R13 7790 z/OS Security Server (RACF) V2 R1 77A0 z/OS Security Server (RACF) V2 R2 77B0 z/OS Security Server (RACF) V2 R3 77C0 z/OS Security Server (RACF) V2 R4 77D0 z/OS Security Server (RACF) V2 R5 77E0 z/OS Security Server (RACF) V3 R1 77F0 z/OS Security Server (RACF) V3 R2
70	46	SMF83SEC	8	EBCDIC	Security label of the user.

Subtype 2 and above

Offset s					
Dec.	Hex.	Name	Length	Format	Description
Security section:					
0	0	SMF83LNK	4	Binary	Value used to link several SMF 83 records to a single event.
4	4	SMF83DES	2	Binary	Descriptor flags Bit Meaning when set 0 The event is a violation 1 User is not defined to RACF 2 Reserved 3 The event is a warning 4 Record contains a version, release, and modification level number (see SMF83VRM) 5 The caller of the R_auditx service indicated always log 6-15 Reserved
6	6	SMF83EVT	1	Binary	Event code.
7	7	SMF83EVQ	1	Binary	Event code qualifier.
8	8	SMF83USR	8	EBCDIC	Identifier of the user associated with this event (jobname is used if the user is not defined to RACF).
16	10	SMF83GRP	8	EBCDIC	Group to which the user was connected (stepname is used if the user is not defined to RACF).
24	18	SMF83REL	2	Binary	Reserved
26	1A	SMF83CNT	2	Binary	Reserved
28	1C	SMF83ATH	1	Binary	Authorities used for processing commands or accessing resources Bit Meaning when set 0-7 Reserved

Offsets					
Dec.	Hex.	Name	Length	Format	Description
29	1D	SMF83REA	1	Binary	Reason for logging. These flags indicate the reason RACF produced the SMF record Bit Meaning when set 0 SETROPTS AUDIT(class) changes to this class of profile are being audited. 1 User being audited 2 SPECIAL users being audited 3 Access to the resource is being audited because of the AUDIT option (specified when profile created or altered by a RACF command), a logging request from the RACROUTE REQUEST=AUTH exit routine, or because the operator granted access during failsoft processing. 4 RACROUTE REQUEST=VERIFY or initACEE failure. 5 This command is always audited 6 Violation detected in command and CMDVIOL is in effect 7 Access to entity being audited because of GLOBALAUDIT option.
30	1E	SMF83TLV	1	Binary	Terminal level number of foreground user (zero if not available).
31	1F	SMF83ERR	1	Binary	Command processing error flag Bit Meaning when set 0 Command had error and RACF could not back out some changes 1 No profile updates were made because of error in RACF processing 2-7 Reserved
32	20	SMF83TRM	8	EBCDIC	Terminal ID of foreground user (zero if not available).
40	28	SMF83JBN	8	EBCDIC	Job name. For RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX records for batch jobs, this field can be zero.
48	30	SMF83RST	4	Binary	Time, in hundredths of a second that the reader recognized the JOB statement for this job for RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX records for batch jobs, this field can be zero.
52	34	SMF83RSD	4	Packed	Date the reader recognized the JOB statement for this job in the form 0cyydddF (where F is the sign) for RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX records for batch jobs, this field can be zero.
56	38	SMF83UID	8	EBCDIC	User identification field from the SMF common exit parameter area. For RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX records for batch jobs, this field can be zero.
64	40	SMF83VER	1	Binary	Version indicator 8 = Version 1, Release 8 or later. As of RACF 1.8.1, SMF83VRM is used instead.

Offset s					
Dec.	Hex.	Name	Length	Format	Description
65	41	SMF83RE2	1	Binary	Additional reasons for logging Bit Meaning when set 0 Security level control for auditing 1 Auditing by LOGOPTIONS 2 Class being audited because of SETROPTS SECLABELAUDIT 3 Class being audited because of SETROPTS COMPATMODE 4 Audited because of SETROPTS APPLAUDIT 5 Audited because user not defined to z/OS UNIX 6 Audited because user does not have appropriate authority for z/OS UNIX 7 Reserved
66	42	SMF83VRM	4	EBCDIC	FMID for RACF
70	46	SMF83SEC	8	EBCDIC	Security Label of the User.
78	4E	SMF83AU2	1	Binary	Authority used continued Bit Meaning when set 0 z/OS UNIX superuser 1 z/OS UNIX system function 2-7 Reserved
79	4F	SMF83RSV	4	Binary	Reserved
80	50	SMF83US2	8	EBCDIC	Identifier of the address space user associated with this event.
88	58	SMF83GR2	8	EBCDIC	Group to which the address space user was connected.

Relocate sections

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

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

The relocate sections for subtype 1 use the standard relocate section format. The data types for the relocate sections for subtype 1 are described in the [“Table of relocate section variable data”](#) on page 56

The relocate sections for subtypes 2 and above use the extended relocate section format. The data types (that is, relocate types) for the subtypes are documented with the product or component that reported the security event. Data type values of 100 and above are reserved for product or component use.

Table 6. RACF SMF record relocate section format

Offsets					
Dec.	Hex.	Name	Length	Format	Description
RACF SMF record standard relocate section format:					
0	0	SMF83DTP	1	Binary	Data type
1	1	SMF83DLN	1	Binary	Length of data that follows.
2	2	SMF83DTA	1-255 (1-FF)	mixed	Data
RACF SMF record extended relocate section format:					
0	0	SMF83TP2	2	Binary	Data type
2	2	SMF83DL2	2	Binary	Length of data that follows.
4	4	SMF83DA2	variable	EBCDIC	Data

The relocate data type values 1-99 that appear in an SMF type 83 subtype 2 or above record are reserved for use by the RACF auditing services. The following table lists those relocate data types that have been assigned. These data types are used only for SMF type 83 subtype 2 records and above.

Table 7. RACF SMF type 83 subtype 2 and above relocates

Data type (SMF83TP2)		Max data length (SMF83DL2)		Format	Audited by event code	Description
Dec.	Hex.	Dec.	Hex.			
1	1	255	FF	EBCDIC	All subtype 2 and above	Subject's distinguished name from the current ACEE
2	2	255	FF	EBCDIC	All subtype 2 and above	Issuers distinguished name from current ACEE
3	3	246	F6	EBCDIC	All subtype 2 and above	Resource name
4	4	8	8	EBCDIC	All subtype 2 and above	Class name
5	5	246	F6	EBCDIC	All subtype 2 and above	Profile name
6	6	7	7	EBCDIC	All subtype 2 and above	FMID of the product requesting event logging
7	7	255	FF	EBCDIC	All subtype 2 and above	Name of the product requesting event logging
8	8	255	FF	EBCDIC	All subtype 2 and above	Log string
9	9	8	8	Binary	All subtype 2 and above	Link value
10	A	510	1FE	EBCDIC	All subtype 2 and above	Authenticated user name
11	B	255	FF	EBCDIC	All subtype 2 and above	Authenticated user registry name
12	C	128	80	EBCDIC	All subtype 2 and above	Authenticated user host name
13	D	16	10	EBCDIC	All subtype 2 and above	Authenticated user authentication mechanism object identifier (OID)
14	E	246	F6	UTF-8	All, except 68, 71, 79, 81, 82, and 85	Authenticated distributed identity user name
15	F	255	FF	UTF-8	All, except 68, 71, 79, 81, 82, and 85	Authenticated distributed identity user registry
100	64	8	8	EBCDIC	Subtype 7	User ID
101	65	20	14	EBCDIC	Subtype 7	Factor name
102	66	255	FF	EBCDIC	Subtype 7	Policy name

Reformatted RACF SMF records

For sorting purposes, the RACF report writer reformats SMF records (types 20, 30, 80, 81 and 83) and uses these reformatted records as input to the modules that produce the RACF reports. If you want to use the RACF report writer exit routine (ICHRSMFE) to produce additional reports or to add additional record selection criteria, you should familiarize yourself with the layouts of these reformatted records.

Record type 20 (job initiation) is written at job initiation (including TSO logon). Record type 30 (common address space work record) is written at normal or abnormal termination of a batch job or step, a TSO session, or a started task. Record type 30 is also written at the expiration of an accounting interval if INTERVAL is specified in SMFPRMxx, at the start of a job (or at the start of the first step after a warm start), and at the expiration of an accounting interval for a system address space. Record types 20 and 30 are documented in *z/OS MVS System Management Facilities (SMF)*.

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

- All records have a common format, independent of which release created them.
- Future changes to the records do not require changes to exit routines (unless you want to process fields that are added in a new release).

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

Reformatted process records

RACF SMF record types 20, 30, 80 and 83 become reformatted process records. These records are variable in length. Note that a RACF SMF record type 80 generated by a SETROPTS or an RVARY command also causes the creation of a reformatted status record.

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

Offsets					
Dec.	Hex.	Name	Length	Format	Description
0	0	RCDLEN	2	binary	Total record length
2	2	-	2	binary	Reserved for IBM's use
4	4	RCDRELNO	1	binary	Release of RACF
5	5	RCDREFMT	1	binary	Reformat indicator (if this byte is X'00', the record has been reformatted to the RACF Version 1 Release 6/7 format)
6	6	RCDSYSID	4	EBCDIC	System identification
10	A	RCDTYPE	1	EBCDIC	Record type (80 decimal)
11	B	RCDTIME	4	packed	Unsigned packed decimal in the form HHMMSSSTH
15	F		1	EBCDIC	Reserved for IBM's use
16	10	RCDDATE	3	packed	Date in form YYDDDF, where F is the sign
19	13	RCDFIXLN	2	binary	Offset from the start of the record to the first relocate section
21	15	RCDCOMLN	2	binary	Offset from the start of the record to the record dependent fields
23	17	RDCNT	2	binary	Number of relocate segments
25	19	RCDEVENT	1	binary	Event code
26	1A	RCDQUAL	1	binary	Event code qualifier

Offsets					
Dec.	Hex.	Name	Length	Format	Description
27	1B	RCD80FLG	1	binary	Descriptor flags: Bit Meaning when set 0 This record is for security violations. 1 This record is for a job/step, not a user/group. 2 This record is truncated. 3 This record is for a warning. 4-7 Reserved for IBM's use.
28	1C		1	binary	Reserved for IBM's use
29	1D	RCDUSER	8	EBCDIC	Identifier of the user for which this event is recorded (or jobname if the user is not defined to RACF)
37	25	RCDGROUP	8	EBCDIC	Group to which the user was connected (or stepname if the user is not defined to RACF)
45	2D	RCDLOGCL	1	binary	Type of event by number: Number Type 1 LOGON/ JOB 2 Entity access 3 RACF command
46	2E	RCDCLASS	8	EBCDIC	Resource class name (see Note 1). This field contains binary zeros for records that are written by the RVARY and SETROPTS commands.
54	36	RCDNAME	44	EBCDIC	Resource name (see Notes 1 and 6). This field contains the user ID for a LOGON/JOB; the resource name for a resource access.
98	62	RCDJOBID	8	EBCDIC	Job name
106	6A		1	EBCDIC	Reserved for IBM's use
107	6B	RCDDATID	3	packed	Date that the reader recognized the JOB card for this job in the form YYDDDF
110	6E	RCDTIMID	4	EBCDIC	Time that the reader recognized the JOB card for this job in the form HHMMSSSTH
114	72	RCDUSRDA	8	EBCDIC	User identification field
122	7A	RCD80TRM	8	EBCDIC	Terminal identification field
130	82	RCD80TML	1	binary	Terminal level number
131	83	RCDOWNER	8	EBCDIC	Owner of the resource
139	8B	RCDUSRSM	20	EBCDIC	User name
159	9F	RCDVRM	4	EBCDIC	Release, version, and modification number
163	A3	RCDSEC	8	EBCDIC	User's security label
171	AB	RCDLINK	4	binary	LINK to connect data sets affected by a security label change with RACF command (ALTDSD, ADDSD, DELDSD) that caused the change.
175	AF	RCDSTYPE	2	binary	SMF record subtype

Reformatted SMF records

Offsets					
Dec.	Hex.	Name	Length	Format	Description
177	B1	RCDNAMEO	2	binary	See Note 6. Offset in variable section to relocate section type if entity name is greater than 44 characters or X'7FFF' if resource name is less than or equal to 44 characters.
179	B3	RCDPVAU1	4	binary	The APPLAUDIT key, part 1 of 2
183	B7	RCDPVAU2	4	binary	The APPLAUDIT key, part 2 of 2

For process records, the record-dependent section is:

Offsets					
Dec.	Hex.	Name	Length	Format	Description
0	0	RCD80ATH	1	binary	Authority used: Bit Meaning when set 0 Normal authority 1 SPECIAL attribute 2 OPERATIONS attribute 3 AUDITOR attribute 4 Exit routine granted authority 5 Failsoft processing 6 Bypassed-user ID=*BYPASS* 7 Trusted attribute
1	1	RCD80REA	2	binary	Reason for logging: Bit Meaning when set 0 Class being audited 1 User being audited 2 Special user being audited 3 Resource being audited, installation-requested logging in effect, or failsoft processing 4 RACINIT failures being audited 5 Command always causes auditing 6 Command violations being audited 7 Audited because GLOBALAUDIT option in effect 8 SECLEVEL audit 9-15 Contains the remaining data from SMF80RE2

Offsets					
Dec.	Hex.	Name	Length	Format	Description
3	3	RCD80ERR	1	binary	Error indicators: Bit Meaning when set 0 Command could not recover 1 Profile not altered 2-7 Reserved for IBM's use
4	4	RCDQUAL1	8	EBCDIC	Qualifier for old data set name (see Note 2)
12	C	RCDQUAL2	8	EBCDIC	Qualifier for new data set name (see Note 3)
20	14	RCDDLEV	1	binary	Data set level number (see Note 4)
21	15	RCDDINT	1	binary	Access authority requested: (see Note 4) Bit Access authority 0 ALTER 1 CONTROL 2 UPDATE 3 READ 4-7 Reserved for IBM's use.
22	16	RCDDALWD	1	binary	Access authority allowed: (see Note 4) Bit Access Authority 0 ALTER 1 CONTROL 2 UPDATE 3 READ 4 NONE 5 EXECUTE 6-7 Reserved for IBM's use
23	17	RCDDVOL	6	EBCDIC	Volume serial (see Note 4)
29	1D	RCDDOLDV	6	EBCDIC	OLDVOL volume serial (see Note 4)
35	23	RCD80GNS	1	binary	1=Generic name specified
36	24	RCD80GSP	1	binary	1=Generic name specified on FROM keyword of PERMIT
37	25	RCD80RRF	1	binary	1=The <i>old</i> name of the RACROUTE REQUEST=DEFINE-renamed data set from data type 33 relocate section
38	26	RCD80RRT	1	binary	1=The <i>new</i> name of the RACROUTE REQUEST=DEFINE-renamed data set from data type 33 relocate section
39	27	RCDGENAM	44	EBCDIC	Generic profile used or generic resource name (see Note 7)

Reformatted SMF records

Offsets					
Dec.	Hex.	Name	Length	Format	Description
83	53	RCDGNNMF	44	EBCDIC	Generic profile used on RACROUTE REQUEST=DEFINE RENAME or generic resource name on RACROUTE REQUEST=DEFINE RENAME Relocate Section: (see Notes 5 and 8)
127	7F	RCDGENAO	2	binary	See Note 7
129	81	RCDGNNMO	2	binary	See Note 8
Variable relocate section map					
+0	+0	RCDDTYPE	1	binary	Data type
+1	+1	RCDDLGT	1	binary	Length of data that follows
+2	+2	RCDDATA	variable	mixed	Data

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

If RCDEVENT is	Resource class name	Resource name
1	USER	user ID (SMF80USR)
2	class name (17)	resource name (1)
3	class name (17)	resource name (1)
4	class name (17)	resource name (1)
5	class name (17)	resource name (1)
6	class name (17)	resource name (1)
7	class name (17)	resource name (1)
8	DATASET	data set name (6)
9	GROUP	group name (6)
10	USER	user ID (6)
11	DATASET	data set name (6)
12	GROUP	group name (6)
13	USER	user ID (6)
14	USER	user ID (6)
15	DATASET	data set name (6)
16	GROUP	group name (6)
17	USER	user ID (6)
18	USER	user ID (6)
19	class name (17)	resource name (9)
20	class name (17)	resource name (9)
21	class name (17)	resource name (9)

If RCDEVENT is	Resource class name	Resource name
22	class name (17)	resource name (9)
23	USER	user ID (6)
24	none	none
25	none	none

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

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

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

Note 5: See “Table of event codes and event code qualifiers” on page 41 and “Table of relocate section variable data” on page 56 earlier in this chapter for a further explanation of these event codes and data types.

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

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

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

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

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

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

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

Field RCDGNNMO contains the offset in the variable section of the reformatted record of relocate type that contains the long resource name.

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

Reformatted status records

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

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

For status records, the record-dependent section is:

Offset s					
Dec.	Hex.	Name	Length	Format	Description
0	00	RCDRACFD	44	EBCDIC	Name of the RACF database for this IPL

Reformatted SMF records

Offset s					
Dec.	Hex.	Name	Length	Format	Description
44	2C	RCDRACFV	6	EBCDIC	Volume identification of RACF database
50	32	RCDRACFU	3	EBCDIC	Unit name of RACF database
53	35	RCD81FLG	1	binary	Options indicators: Bit Meaning when set 0 No RACROUTE REQUEST=VERIFY statistics are recorded 1 No DATASET statistics are recorded 2 RACROUTE REQUEST=VERIFY preprocessing exit routine, ICHRIX01, is active 3 RACROUTE REQUEST=AUTH preprocessing exit routine, ICHRCX01, is active 4 RACROUTE REQUEST=DEFINE preprocessing exit routine, ICHRDY01, is active 5 RACROUTE REQUEST=VERIFY postprocessing exit routine, ICHRIX02, is active 6 RACROUTE REQUEST=AUTH postprocessing exit routine, ICHRCX02, is active 7 New-password exit routine, ICHPWX01, is active
54	36	RCDUVOL	6	EBCDIC	Volume identification of UADS data set
60	3C	RCDUDSN	44	EBCDIC	Data set name of the UADS data set for this IPL
104	68	RCD81FG2	1	binary	Options indicators: Bit Meaning when set 0 No tape volume statistics are recorded 1 No DASD volume statistics are recorded 2 No terminal statistics are recorded 3 Command exit routine ICHCNX00 is active 4 Command exit routine ICHCCX00 is active 5 ADSP is not active 6 Encryption exit routine, ICHDEX01, is active 7 Naming convention table, ICHNCV00, is present

Offset s					
Dec.	Hex.	Name	Length	Format	Description
105	69	RCD81OP3	1	binary	Options indicators: Bit Meaning when set 0 Tape volume protection in effect 1 No duplicate data set protection in effect 2 DASD volume protection in effect 3 Reserved for IBM's use 4 RACROUTE REQUEST=FASTAUTH preprocessing exit routine (ICHRFX01) is active 5 RACROUTE REQUEST=LIST pre- and postprocessing exit routine is active 6 RACROUTE REQUEST=LIST selection exit routine is active 7 RACROUTE REQUEST=DEFINE postprocessing exit routine is active
106	6A	RCD81AOP	1	binary	Options indicators: Bit Meaning when set 0 Log all users 1 Log all groups 2 Log data set class 3 Log tape volume class 4 Log DASD volume class 5 Log terminal class 6 Log command violations 7 Log special users

Reformatted SMF records

Offset s					
Dec.	Hex.	Name	Length	Format	Description
107	6B	RCD81TMO	1	binary	Options indicators: Bit Meaning when set 0 Terminal authorization checking in effect 1 UACC for undefined terminals is NONE 2 REALDSN is in effect 3 JES-XBMALLRACF is in effect 4 JES-EARLYVERIFY is in effect 5 JES-BATCHALLRACF is in effect 6 RACROUTE REQUEST=FASTAUTH postprocessing exit is active 7 Reserved for IBM's use
108	6C	RCD81PIV	1	binary	Maximum password interval
109	6D	RCD81MFG	1	binary	Model flags: Bit Meaning when set 0 ModelGDG 1 ModelUSER 2 ModelGROUP 3-7 Reserved for IBM's use
110	6E	RCD81MSF	1	binary	Miscellaneous processing flags: Bit Meaning when set 0 GRPLIST active 1 Generic profile checking in effect for data set 2 GENCMD in effect for data set class 3 ADSP attribute bypassed 4-7 Reserved for IBM's use

Offset s					
Dec.	Hex.	Name	Length	Format	Description
111	6F	RCD81IFG	1	binary	Internal processing flags: Bit Meaning when set 0 The SETROPTS command caused RACFRW to generate this record 1 The RVARY command caused RACFRW to generate this record 2 RACF was varied active by RVARY command 3 This record is incomplete (truncated) 4 RVARY SWITCH was issued 5-7 Reserved for IBM's use
112	70	RCD81QL	8	char	Single level data set name prefix
120	78	RCD81A02	1	binary	Options indicator Bit Meaning when set 0 Log OPERATIONS user 1-7 Reserved for IBM's use 2 RACF was varied active by RVARY command.
121	79	RCD81OP4	1	binary	Options indicators Bit Meaning when set 0 Tape DSN active 1 PROTECTALL active 2 PROTECTALL warning 3 Erase-on-scratch 4 Erase by SECLEVEL 5 Erase all files 6-7 Reserved for IBM's use

Reformatted SMF records

Offset s					
Dec.	Hex.	Name	Length	Format	Description
122	7A	RCD81OP5	1	binary	Options indicators Bit Meaning when set 0 Program control active 1 ACEE compression/expansion exit active 2 RACROUTE REQUEST=FASTAUTH postprocessing exit ICHRFX04 active 3 RACROUTE REQUEST=FASTAUTH postprocessing exit ICHRFX04 active 4-7 Reserved for IBM's use
124	7C	RCD81RPD	2	binary	Data set retention period
126	7E	RCD81SLV	1	char	SECLEVEL number
127	7F	RCD81SLC	1	binary	SECLEVEL for auditing number
128	80	RCD81BOP	1	binary	B1 security options Bit Meaning when set 0 SECLABELCONTROL active 1 CATDSNS active 2 MLQUIET active 3 MLSTABLE active 4 MLS active 5 MLACTIVE active 6 GENERICOWNER active 7 SECLABELAUDIT active
129	81	RCD81SIN	2	binary	SESSION INTERVAL
131	83	RCD81SYS	8	char	User ID for JES SYSOUTNAME
139	8B	RCD81UND	8	char	User ID for JES undefined user

Offset s					
Dec.	Hex.	Name	Length	Format	Description
147	93	RCD81BOX	1	binary	B1 security options extension byte Bit Meaning when set 0 COMPATMODE 1 CATDSNS failures 2 MLS failures 3 MLACTIVE failures 4-7 Reserved for IBM's use
148	94	RCD81PRI	3	EBCDIC	Primary language default
151	97	RCD81SEC	3	EBCDIC	Secondary language default
Variable relocate section map					
+0	+0	RCDDTYPE	1	binary	Data type
+1	+1	RCDDLGT	1	binary	Length of data that follows
+2	+2	RCDDATA	variable	mixed	Data

Note: Only data types (SMF80DTP) 21, 30, 32, 34, 35 or 36 are generated for a reformatted status record. See [“Table of relocate section variable data”](#) on [page 56](#) for a further explanation of these data types.

Chapter 6. The format of the unloaded SMF type 80 data

The SMF data unload utility can unload SMF data in two formats:

- A tabular format, suitable for export to a relational database manager. This topic documents that format.
- An eXtended Markup Language (XML) document, which can be rendered into different formats such as Web pages (HTML). An installation can write applications that interpret the data to generate custom reports. For information about how to convert the field names in the tabular format to XML tags, see [“XML grammar” on page 170](#).

IRRADU00 record format

The following sections contain a detailed description of the records that are produced by the RACF SMF data unload utility. The output of the utility is a series of records that represents the security relevant SMF data that is the input to the utility. These records are in a format suitable for export to the relational data manager of an installation's choice.

Each record that is produced consists of two parts:

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

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

1. Descriptive name for the field
2. Type of field

Char

Character data

Integer

EBCDIC numeric data

Time

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

Date

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

Yes/No

Flag data, having the value YES or NO

3. Starting position for the field
4. Ending position for the field
5. Free-form description of the field, which may contain the valid value constraints.

In some cases, the input SMF record does not contain all of the data that are indicated in the output record mappings shown in the following sections. In these cases, IRRADU00 places blanks in the fields.

For the audit records created for RACF commands, the exact order and format of the unloaded keywords and operands from the commands (contained within the fields whose names end with "_SPECIFIED", "_IGNORED", and "_FAILED") are not part of the programming interface.

Furthermore, for RACF commands that allow segment fields to be specified (ADDUSER, ALTUSER, ADDGROUP, ALTGROU, ADDSD, ALTDSD, RDEFINE, and RALTER), only the keywords that correspond to the base segment in the RACF database appear in the SMF unload fields whose names end

with "_IGNORED" and "_FAILED". Keywords that correspond to segment fields in the RACF database, such as "TSO(ACCTNUM(1234))" or "SESSION(INTERVAL(20))" appear in fields whose names end with "_SPECIFIED", even if the segment keywords fail because of field level access checking.

XML grammar

The RACF SMF data unload utility can generate an eXtensible Markup Language (XML) document that contains the SMF data. The names of the tags and the syntax of the values are defined in an XML schema document, which is used to validate the data that is contained in an instance document. The RACF schema document is provided by IBM in SYS1.SAMPLIB, in member IRRSCHEM.

In general, the RACF XML tag names are derived from the field names in the tabular output and the field values are not altered. Therefore, if you use this output, be aware that fields such as logstr, workattr, name, and others, can be set by any user. If so, the fields are passed unaltered into the XML output. For example, if the XML output includes HTML tags and the output is saved unprocessed in an HTML file, those HTML tags are processed as-is and might generate unexpected results.

Steps for converting RACF field names to XML tag names

About this task

Before you begin: You need to know the name of the RACF field name that you want to convert.

Perform the following steps to convert a RACF field name from the tabular format produced by the RACF SMF data unload utility to an XML tag name.

Procedure

1. Remove the column name and the first "_" character from the field name.

2. Capitalize the first character after each remaining "_" character in the field name. Change all other characters to lowercase.

3. Remove all remaining "_" characters.

Results

When you are done, you have the name of the XML tag that corresponds with the field name that you started with.

Exceptions to this procedure are:

Field name	XML tag name
RINI_TERM	riniTerm
SECL_LINK	link
CAUD_REQUEST_WRITE	caudRequestWrite
CAUD_REQUEST_READ	caudRequestRead
CAUD_REQUEST_EXEC	caudRequestExec
SSCL_OLDSECL	oldSecl
<col>_logstring	logstr
KTKT_PRINCIPAL	kerbPrincipal

Field name	XML tag name
PDAC_PRINCIPAL	pdasPrincipal
any field with RESERVED in the name	These fields have no XML tag.
ACC_NAME	profileName
APPC_NAME	profileName

Example: Converting the field name INIT_USER_NAME to an XML tag name:

1. Start with INIT_USER_NAME

```
INIT_USER_NAME
```

2. Remove the column name (INIT) and the first "_" character from the field name.

```
USER_NAME
```

3. Capitalize the first character after each remaining "_" character in the field name. Change all other characters to lowercase.

```
user_Name
```

4. Remove all remaining "_" characters.

```
userName
```

The format of the header portion of the unloaded SMF type 30 and type 80

Table 8 on page 172 describes the format of the header portion of the record. RACF constructs the header portion of the record from the SMF record. Because each of the SMF record types that IRRADU00 processes contain different data, some fields of the header portion of the unloaded SMF record contain blanks. For example, JOBINIT records that are created from type 30 SMF records have blanks for these fields:

- INIT_VIOLATION
- INIT_USR_NDFND
- INIT_USER_WARNING
- All of the INIT_AUTH_ fields
- All of the INIT_LOG_ fields
- INIT_TERM_LEVEL
- INIT_BACKOUT_FAIL
- INIT_PROF_SAME
- INIT_TERM
- INIT_READ_TIME
- INIT_READ_DATE
- INIT_USR_SECL
- INIT_RACF_VERSION

The <col_id> string is replaced by the column identifier for each record created. See Table 9 on page 174 for a list of the valid column identifiers.

Table 8. Format of the header portion of the unloaded SMF records					
Field name	Type	Length	Position		Comments
			Start	End	
<col_id>_EVENT_TYPE	Char	8	1	8	Type of event that is described. Valid values are shown in Table 9 on page 174. A numeric value indicates that the event code was not translated. Only header information is created for records that have an untranslated event code.
<col_id>_EVENT_QUAL	Char	8	10	17	A qualification of the type of event that is being described. Valid values are shown in the tables that accompany each of the record extension descriptions.
<col_id>_TIME_WRITTEN	Time	8	19	26	Time that the record was written to SMF.
<col_id>_DATE_WRITTEN	Date	10	28	37	Date that the record was written to SMF.
<col_id>_SYSTEM_SMFID	Char	4	39	42	SMF system ID of the system from which the record originates.
<col_id>_VIOLATION	Yes/No	4	44	47	Does this record represent a violation?
<col_id>_USER_NDFND	Yes/No	4	49	52	Was this user not defined to RACF?
<col_id>_USER_WARNING	Yes/No	4	54	57	Was this record created because of WARNING?
<col_id>_EVT_USER_ID	Char	8	59	66	User ID associated with the event.
<col_id>_EVT_GRP_ID	Char	8	68	75	Group name associated with the event.
<col_id>_AUTH_NORMAL	Yes/No	4	77	80	Was normal authority checking a reason for access being allowed?
<col_id>_AUTH_SPECIAL	Yes/No	4	82	85	Was special authority checking a reason for access being allowed?
<col_id>_AUTH_OPER	Yes/No	4	87	90	Was operations authority checking a reason for access being allowed?
<col_id>_AUTH_AUDIT	Yes/No	4	92	95	Was auditor authority checking a reason for access being allowed?
<col_id>_AUTH_EXIT	Yes/No	4	97	100	Was exit checking a reason for access being allowed?
<col_id>_AUTH_FAILSFT	Yes/No	4	102	105	Was failsoft checking a reason for access being allowed?
<col_id>_AUTH_BYPASS	Yes/No	4	107	110	Was the use of the user ID *BYPASS* a reason for access being allowed?
<col_id>_AUTH_TRUSTED	Yes/No	4	112	115	Was trusted authority checking a reason for access being allowed?
<col_id>_LOG_CLASS	Yes/No	4	117	120	Was SETR AUDIT(class) checking a reason for this event to be recorded? For Event Code 1, the class is USER, and a value of YES indicates that a password or password phrase was changed during logon.
<col_id>_LOG_USER	Yes/No	4	122	125	Was auditing requested for this user?
<col_id>_LOG_SPECIAL	Yes/No	4	127	130	Was auditing requested for access granted due to the SPECIAL or OPERATIONS privilege? To determine whether SPECIAL or OPERATIONS authority was used, see <col_id>_AUTH_SPECIAL and <col_id>_AUTH_OPER.
<col_id>_LOG_ACCESS	Yes/No	4	132	135	Did the profile or UNIX file indicate audit, or did FAILSOFT processing allow access, or did the RACHECK exit indicate auditing?
<col_id>_LOG_RACINIT	Yes/No	4	137	140	Did the RACINIT fail?
<col_id>_LOG_ALWAYS	Yes/No	4	142	145	Is this command always audited?
<col_id>_LOG_CMDVIOL	Yes/No	4	147	150	Was this event audited due to CMDVIOL?

Table 8. Format of the header portion of the unloaded SMF records (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
<col_id>_LOG_GLOBAL	Yes/No	4	152	155	Was this event audited due to GLOBALAUDIT, or, for UNIX files, the AUDITOR audit bits?
<col_id>_TERM_LEVEL	Integer	3	157	159	The terminal level associated with this audit record.
<col_id>_BACKOUT_FAIL	Yes/No	4	161	164	Did RACF fail in backing out the data?
<col_id>_PROF_SAME	Yes/No	4	166	169	Did a RACF error cause the profile to not be changed?
<col_id>_TERM	Char	8	171	178	The terminal associated with the event.
<col_id>_JOB_NAME	Char	8	180	187	The job name associated with the event.
<col_id>_READ_TIME	Time	8	189	196	The time that the job entered the system.
<col_id>_READ_DATE	Date	10	198	207	The date that the job entered the system.
<col_id>_SMF_USER_ID	Char	8	209	216	User ID from SMF common area. This value is managed by SMF and the SMF processing exits.
<col_id>_LOG_LEVEL	Yes/No	4	218	221	Was this event audited due to SECLEVEL auditing?
<col_id>_LOG_VMEVENT	Yes/No	4	223	226	Was this event audited due to VMEVENT auditing?
<col_id>_LOG_LOGOPT	Yes/No	4	228	231	Was this event audited due to SETR LOGOPTIONS auditing?
<col_id>_LOG_SECL	Yes/No	4	233	236	Was this event audited due to SETR SECLABELAUDIT auditing?
<col_id>_LOG_COMPATM	Yes/No	4	238	241	Was this event audited due to SETR COMPATMODE auditing?
<col_id>_LOG_APPLAUD	Yes/No	4	243	246	Was this event audited due to SETR APPLAUDIT?
<col_id>_LOG_NONOMVS	Yes/No	4	248	251	Did this user try to use z/OS UNIX without being defined as a z/OS UNIX user (that is, is the user's OMVS segment in the RACF database missing)?
<col_id>_LOG_OMVSNPRV	Yes/No	4	253	256	The service that was requested requires that the user be the z/OS UNIX super-user.
<col_id>_AUTH_OMVSSU	Yes/No	4	258	261	Was the z/OS UNIX superuser authority used to grant the request?
<col_id>_AUTH_OMVSSYS	Yes/No	4	263	266	Was the request granted because the requestor was z/OS UNIX itself?
<col_id>_USR_SECL	Char	8	268	275	The security label associated with this user.
<col_id>_RACF_VERSION	Char	4	277	280	The version of RACF on the system that audited the event.

Note: An ACEE is unlikely to have additional identity context information for many events, such as RACF commands.

Event codes

The RACF SMF data unload utility creates records that represent the audit information for each type of auditable event. [Table 9 on page 174](#) contains a list of all of the supported event codes.

COLUMN NAME DESCRIPTION

Event Code Name

Name of the event code

Column ID (Col ID)

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

Event Code

The number assigned to this event code by RACF

Description

A description of the event code.

Where Described

Where you can find the record definitions.

Table 9. Event codes and descriptions

Event code name	Col ID	Event code	Description	Where described
JOBINIT	INIT	01	Job initiation	“The JOBINIT record extension” on page 177
ACCESS	ACC	02	Resource access, other than file or directory.	“The ACCESS record extension” on page 182
ADDVOL	ADV	03	ADDVOL/CHGVOL	“The ADDVOL record extension” on page 185
RENAMEDS	REN	04	Rename data set, SFS file, or SFS directory	“The RENAMEDS record extension” on page 187
DELRES	DELR	05	Delete resource	“The DELRES record extension” on page 189
DELVOL	DELV	06	Delete volume	“The DELVOL record extension” on page 191
DEFINE	DEF	07	Define resource	“The DEFINE record extension” on page 192
ADDSD	AD	08	ADDSD command	“The ADDSD record extension” on page 194
ADDGROUP	AG	09	ADDGROUP command	“The ADDGROUP record extension” on page 195
ADDUSER	AU	10	ADDUSER command	“The ADDUSER record extension” on page 197
ALTDSD	ALD	11	ALTDSD command	“The ALTDSD record extension” on page 199
ALTGROUP	ALG	12	ALTGROUP command	“The ALTGROUP record extension” on page 200
ALTUSER	ALU	13	ALTUSER command	“The ALTUSER record extension” on page 202
CONNECT	CON	14	CONNECT command	“The CONNECT record extension” on page 203
DELDSD	DELD	15	DELDSD command	“The DELDSD record extension” on page 205
DELGROUP	DELG	16	DELGROUP command	“The DELGROUP record extension” on page 206
DELUSER	DELU	17	DELUSER command	“The DELUSER record extension” on page 208
PASSWORD	PWD	18	PASSWORD command	“The PASSWORD record extension” on page 209
PERMIT	PERM	19	PERMIT command	“The PERMIT record extension” on page 211
RALTER	RALT	20	RALTER command	“The RALTER record extension” on page 212
RDEFINE	RDEF	21	RDEFINE command	“The RDEFINE record extension” on page 214
RDELETE	RDEL	22	RDELETE command	“The RDELETE record extension” on page 215
REMOVE	REM	23	REMOVE command	“The REMOVE record extension” on page 217
SETROPTS	SETR	24	SETROPTS command	“The SETROPTS record extension” on page 218
RVARY	RVAR	25	RVARY command	“The RVARY record extension” on page 220
APPCLU	APPC	26	APPC session	“The APPCLU record extension” on page 221
GENERAL	GEN	27	General purpose	“The general event record extension” on page 223
DIRSRCH	DSCH	28	Directory Search	“The directory search record extension” on page 224
DACCESS	DACC	29	Check access to a directory	“The check directory access record extension” on page 227
FACCESS	FACC	30	Check access to file	“The check file access record extension” on page 229
CHAUDIT	CAUD	31	Change audit options	“The change audit record extension” on page 231
CHDIR	CDIR	32	Change current directory	“The change directory record extension” on page 234

Table 9. Event codes and descriptions (continued)

Event code name	Col ID	Event code	Description	Where described
CHMOD	CMOD	33	Change file mode	“The change file mode record extension” on page 236
CHOWN	COWN	34	Change file ownership	“The change file ownership record extension” on page 239
CLRSETID	CSID	35	Clear SETID bits for a file	“The clear SETID bits record extension” on page 241
EXESETID	ESID	36	EXEC with SETUID/SETGID	“The EXEC SETUID/SETGID record extension” on page 243
GETPSENT	GPST	37	Get z/OS UNIX process entry	“The GETPSENT record extension” on page 245
INITOEDP	IOEP	38	Initialize z/OS UNIX process	“The initialize z/OS UNIX record extension” on page 247
TERMOEDP	TOEP	39	z/OS UNIX process complete	“The z/OS UNIX process completion record” on page 249
KILL	KILL	40	Terminate a process	“The KILL record extension” on page 250
LINK	LINK	41	LINK	“The LINK record extension” on page 252
MKDIR	MDIR	42	Make directory	“The MKDIR record extension” on page 254
MKNOD	MNOD	43	Make node	“The MKNOD record extension” on page 257
MNTFSYS	MFS	44	Mount a file system	“The mount file system record extension” on page 260
OPENFILE	OPEN	45	Open a new file	“The OPENFILE record extension” on page 262
PTRACE	PTRC	46	PTRACE authority checking	“The PTRACE record extension” on page 265
RENAMEF	RENF	47	Rename file	“The rename file record extension” on page 267
RMDIR	RDIR	48	Remove directory	“The RMDIR record extension” on page 269
SETEGID	SEGI	49	Set effective z/OS UNIX group identifier (GID).	“The SETEGID record extension” on page 271
SETEUID	SEUI	50	Set effective z/OS UNIX user identifier (UID)	“The SETEUID record extension” on page 273
SETGID	SGI	51	Set z/OS UNIX group identifier (GID).	“The SETGID record extension” on page 275
SETUID	SUI	52	Set z/OS UNIX user identifier (UID)	“The SETUID record extension” on page 276
SYMLINK	SYML	53	SYMLINK	“The SYMLINK record extension” on page 278
UNLINK	UNL	54	UNLINK	“The UNLINK record extension” on page 280
UMNTFSYS	UFS	55	Unmount file system	“The unmount file system record extension” on page 282
CHKFOWN	CFOW	56	Check file owner	“The check file owner record extension” on page 284
CHKPRIV	CPRV	57	Check privilege	“The check privilege record extension” on page 286
OPENSTTY	OSTY	58	Open subsidiary TTY	“The open subsidiary TTY record extension” on page 287
RACLINK	RACL	59	RACLINK command	“The RACLINK command record extension” on page 289
IPCCHK	ICLK	60	Check IPC access	“The IPCCHK record extension” on page 292
IPCGET	IGET	61	IPCGET	“The IPCGET record extension” on page 294
IPCCTL	ICTL	62	IPCCTL	“The IPCCTL record extension” on page 296
SETGROUP	SETG	63	SETGROUP	“The SETGROUP record extension” on page 299
CKOWN2	CKO2	64	CKOWN2	“The CKOWN2 record extension” on page 301

Table 9. Event codes and descriptions (continued)				
Event code name	Col ID	Event code	Description	Where described
R_AUDIT	ACCR	65	Access Rights	“The access rights record extension” on page 303
RACDCERT	RACD	66	RACDCERT command	“The RACDCERT command record extension” on page 304
INITACEE	INTA	67	InitACEE	“The InitACEE record extension” on page 306
KTICKET	KTKT	68	Grant of initial Kerberos ticket	“The Network Authentication Service record extension” on page 308
RPKIGENC	RPKG	69	Certificate GENCERT request	“The RPKIGENC record extension” on page 308
RPKIEXPT	RPKE	70	Certificate EXPORT request	“The RPKIEXPT record extension” on page 310
PDACCESS	PDAC	71	Policy Director Authorization Services access control decision	“The Policy Director Authorization Services record extension” on page 312
RPKIREAD	RPKR	72	Certificate administration - read record	“The RPKIREAD record extension” on page 313
RPKIUPDR	RPKU	73	Certificate administration - update request record	“The RPKIUPDR record extension” on page 315
RPKIUPDC	RPKC	74	Certificate administration - update certificate record	“The RPKIUPDC record extension” on page 317
SETFACL	SACL	75	ACL entry changes	“The SETFACL record extension” on page 318
DELFACL	DACL	76	ACL deletion	“The DELFACL record extension” on page 321
SETFSECL	SSCL	77	Set security label of a z/OS UNIX file	“The SETFSECL record extension” on page 323
WRITEDWN	WDWN	78	Set write-down privilege	“The WRITEDWN record extension” on page 324
PKIDPUBR	PKDP	79	CRL publication	“The PKIDPUBR record extension” on page 326
RPKIRESP	RPKO	80	Created by RPKIRESP	“The RPKIRESP record extension” on page 326
PTEVAL	PTEV	81	PassTicket evaluation	“The PassTicket evaluation (PTEVAL) record extension” on page 327
PTCREATE	PTCR	82	PassTicket generation	“The PassTicket generation (PTCREATE) record extension” on page 329
RPKISCEP	RPKS	83	R_PKIServ SCEPREQ	“The RPKISCEP record extension” on page 331
RDATAUPD	RPUT	84	R_datalib write function	“The RDATAUPD record extension” on page 333
PKIAURNW	PKRN	85	R_PKIServ certificate renewal	“The PKIAURNW record extension” on page 335
PGMVERYF	PGMV	86	R_PgmSignVer signature verification	“The PGMVERYF record extension” on page 336
RACMAP	RACM	87	Defines the association between the distributed user identity and a RACF defined user ID	“The RACMAP record extension” on page 337
AUTOPROF	AUTO	88	Logs events that automatically modify RACF profiles	“The AUTOPROF record extension” on page 339
RPKIQREC	RPKQ	89	Finds a list of certificates of which have key pairs generated by PKI Services using the specified requester's email address and pass phrase	“The RPKIQREC record extension” on page 341
PKIGENC		90	GENCERT request	“The PKIGENC record extension” on page 342

Table 9. Event codes and descriptions (continued)

Event code name	Col ID	Event code	Description	Where described
PRLIMIT		91	prlimit() API	“The PRLIMIT record extension” on page 343

Record extensions

The following topics describe event-specific information. The extensions reflect the relocate section data for a specific event code. Fields in the event-specific information might contain blanks because not all relocate sections are created for a given event code.

The JOBINIT record extension

Table 10 on page 177 describes the format of a record that is created by the RACINIT function, which occurs for user logons, batch job initiations, and at other times during the life of a unit of work. These fields are only present on JOBINIT records that are created from SMF type 80 records. JOBINIT records that are created from SMF type 30 records contain blanks in these fields.

The event qualifiers that can be associated with a JOBINIT event are shown in [Table 11 on page 180](#).

Table 10. Format of the job initiation record extension (event code 01)

Field name	Type	Length	Position		Comments
			Start	End	
INIT_APPL	Char	8	282	289	Application name specified on the REQUEST=VERIFY.
INIT_LOGSTR	Char	255	291	545	LOGSTR= data from the RACROUTE
INIT_BAD_JOBNAME	Char	8	547	554	The invalid job name that was processed.
INIT_USER_NAME	Char	20	556	575	The name associated with the user ID.
INIT_UTK_ENCR	Yes/No	4	577	580	Is the UTOKEN associated with this user encrypted?
INIT_UTK_PRE19	Yes/No	4	582	585	Is this a pre-1.9 token?
INIT_UTK_VERPROF	Yes/No	4	587	590	Is the VERIFYX propagation flag set?
INIT_UTK_NJEUNUSR	Yes/No	4	592	595	Is this the NJE undefined user?
INIT_UTK_LOGUSR	Yes/No	4	597	600	Is UAUDIT specified for this user?
INIT_UTK_SPECIAL	Yes/No	4	602	605	Is this a SPECIAL user?
INIT_UTK_DEFAULT	Yes/No	4	607	610	Is this a default token?
INIT_UTK_UNKNUSR	Yes/No	4	612	615	Is this an undefined user?
INIT_UTK_ERROR	Yes/No	4	617	620	Is this user token in error?
INIT_UTK_TRUSTED	Yes/No	4	622	625	Is this user a part of the trusted computing base (TCB)?
INIT_UTK_SESSTYPE	Char	8	627	634	The session type of this session. See z/OS Security Server RACROUTE Macro Reference for a description of the valid values for session type. A null session type results in the unloading of blanks.
INIT_UTK_SURROGAT	Yes/No	4	636	639	Is this a surrogate user?
INIT_UTK_REMOTE	Yes/No	4	641	644	Is this a remote job?
INIT_UTK_PRIV	Yes/No	4	646	649	Is this a privileged user ID?
INIT_UTK_SECL	Char	8	651	658	The security label of the user.
INIT_UTK_EXECNODE	Char	8	660	667	The execution node of the work.
INIT_UTK_SUSER_ID	Char	8	669	676	The submitting user ID.
INIT_UTK_SNODE	Char	8	678	685	The submitting node.

Table 10. Format of the job initiation record extension (event code 01) (continued)

Field name	Type	Length	Position		Comments
			Start	End	
INIT_UTK_SGRP_ID	Char	8	687	694	The submitting group name.
INIT_UTK_SPOE	Char	8	696	703	The port of entry.
INIT_UTK_SPCCLASS	Char	8	705	712	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
INIT_UTK_USER_ID	Char	8	714	721	User ID associated with the record.
INIT_UTK_GRP_ID	Char	8	723	730	Group name associated with the record.
INIT_UTK_DFT_GRP	Yes/No	4	732	735	Is a default group assigned?
INIT_UTK_DFT_SECL	Yes/No	4	737	740	Is a default security label assigned?
INIT_APPC_LINK	Char	16	742	757	A key to link together audit record together for a user's APPC transaction processing work.
INIT_UTK_NETW	Char	8	759	766	The port of entry network name.
INIT_RES_NAME	Char	255	768	1022	Resource name.
INIT_CLASS	Char	8	1024	1031	Class name.
INIT_X500_SUBJECT	Char	255	1033	1287	Subject's name associated with this event.
INIT_X500_ISSUER	Char	255	1289	1543	Issuer's name associated with this event.
INIT_SERVSECL	Char	8	1545	1552	Security label of the server.
INIT_SERV_POENAME	Char	64	1554	1617	SERVAUTH resource or profile name.
INIT_CTX_USER	Char	510	1619	2128	Authenticated user name.
INIT_CTX_REG	Char	255	2130	2384	Authenticated user registry name.
INIT_CTX_HOST	Char	128	2386	2513	Authenticated user host name.
INIT_CTX_MECH	Char	16	2515	2530	Authenticated user authentication mechanism object identifier (OID).
INIT_IDID_USER	Char	985	2532	3516	Authenticated distributed user name.
INIT_IDID_REG	Char	1021	3518	4538	Authenticated distributed user registry name.
INIT_ACEE_VLF	Yes/No	4	4540	4543	The ACEE was created from the VLF cache.
INIT_MFA_USER	Yes/No	4	4545	4548	The user has active MFA factors.
INIT_MFA_FALLBACK	Yes/No	4	4550	4553	The MFA user is allowed to fall back to password authentication when IBM MFA is unavailable.
INIT_MFA_UNAVAIL	Yes/No	4	4555	4558	MFA was unavailable to make an authentication decision for the IBM MFA user.
INIT_MFA_PWD_EXPIRED	Yes/No	4	4560	4563	IBM MFA requested that RACROUTE REQUEST=VERIFY return the password-expired return code.
INIT_MFA_NPWD_INV	Yes/No	4	4565	4568	IBM MFA requested that RACROUTE REQUEST=VERIFY return the new-password-invalid return code.
INIT_MFA_PART_SUCC	Yes/No	4	4570	4573	IBM MFA requested that RACROUTE REQUEST=VERIFY return the password-invalid return code, but not to increment the password revoke count (partial success - needs more information).
INIT_RELO443_EXTENDED	Yes/No	4	4575	4578	Relocate 443 is extended up to field INIT_SERVICE_RSNC. When this bit is off, the last field is INIT_AUTH_RSN2.
INIT_PASSWORD_EVAL	Yes/No	4	4580	4583	The supplied password was evaluated.

Table 10. Format of the job initiation record extension (event code 01) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
INIT_PASSWORD_SUCC	Yes/No	4	4585	4588	The supplied password was evaluated successfully.
INIT_PHRASE_EVAL	Yes/No	4	4590	4593	The supplied password phrase was evaluated.
INIT_PHRASE_SUCC	Yes/No	4	4595	4598	The supplied password phrase was evaluated successfully.
INIT_PASSTICKET_EVAL	Yes/No	4	4600	4603	The supplied password was evaluated as a PassTicket.
INIT_PASSTICKET_SUCC	Yes/No	4	4605	4608	The supplied password was evaluated successfully as a PassTicket.
INIT_MFA_SUCC	Yes/No	4	4610	4613	MFA authentication successful.
INIT_MFA_FAIL	Yes/No	4	4615	4618	MFA authentication unsuccessful.
INIT_AUTH_RSN1	Char	8	4620	4627	MFA Authentication return code. Expressed as hexadecimal number.
INIT_AUTH_RSN2	Char	8	4629	4636	MFA Authentication reason code. Expressed as hexadecimal number.
INIT_AUTH_RSN3	Char	8	4638	4645	PassTicket Authentication return code. Expressed as hexadecimal number.
INIT_AUTH_RSN4	Char	8	4647	4654	PassTicket Authentication reason code. Expressed as hexadecimal number.
INIT_PWD_PHR_EXPIRED	Yes/No	1	4656	4659	The supplied password or password phrase was expired.
INIT_NPWD_NPHR_NONVAL	Yes/No	1	4661	4664	The supplied new password or new password phrase was not valid.
INIT_IDT_EVAL	Yes/No	1	4666	4669	The supplied Identity Token (IDT) was evaluated.
INIT_IDT_SUCC	Yes/No	1	4671	4674	The supplied Identity Token (IDT) was evaluated successfully.
INIT_MFA_REAUTHENT	Yes/No	1	4676	4679	IBM MFA requested that RACROUTE REQUEST=VERIFY return the password-invalid return code, but not to increment the password revoke count (reauthentication requested).
INIT_LPT_EVAL	Yes/No	4	4681	4684	The supplied Password was evaluated as a legacy PassTicket.
INIT_LPT_SUCC	Yes/No	4	4686	4689	The supplied Password was evaluated successfully as a legacy PassTicket.
INIT_EPT_UPPER_EVAL	Yes/No	4	4691	4694	The supplied Password was evaluated as an enhanced PassTicket type UPPER.
INIT_EPT_UPPER_SUCC	Yes/No	4	4696	4699	The supplied Password was evaluated successfully as an enhanced PassTicket type UPPER.
INIT_EPT_MIXED_EVAL	Yes/No	4	4701	4704	The supplied Password was evaluated as an enhanced PassTicket type MIXED.
INIT_EPT_MIXED_SUCC	Yes/No	4	4706	4709	The supplied Password was evaluated successfully as an enhanced PassTicket type MIXED.
INIT_IDT_FROM_SEC_ENV	Yes/No	4	4711	4714	IDT from existing security environment.
INIT_RELO443_EXTEND_2	Yes/No	4	4716	4719	Relocate 443 is extended up to field INIT_RESERVED_22.
INIT_RESERVED_09	Yes/No	4	4721	4724	Reserved for IBM use.
INIT_RESERVED_10	Yes/No	4	4726	4729	Reserved for IBM use.
INIT_RESERVED_11	Yes/No	4	4731	4734	Reserved for IBM use.

Table 10. Format of the job initiation record extension (event code 01) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
INIT_DERIVED_APPL_NAM	Char	8	4736	4743	Derived Application Name
INIT_IDT_VALIDTN_RSNC	Char	8	4745	4752	IDT Validation Reason Code.
INIT_IDT_ERROR_RSNC	Char	8	4754	4761	IDT Error Reason Code.
INIT_SERVICE_CODE	Char	8	4763	4770	Failing Service Identifier.
INIT_SERVICE_RC	Char	8	4772	4779	Failing Service Return Code.
INIT_SERVICE_RSNC	Char	8	4781	4788	Failing Service Reason Code.
INIT_IDT_SIG_ALG	Char	10	4790	4799	Signature algorithm from IDT.
INIT_IDT_KID	Char	32	4801	4832	Key Identifier from IDT.
INIT_RESERVED_12	Char	100	4834	4933	Reserved for IBM use.
INIT_RESERVED_13	Char	100	4935	5034	Reserved for IBM use.
INIT_RESERVED_14	Char	246	5036	5281	Reserved for IBM use.
INIT_IDT_SIG_EVAL_PRI	Yes/No	4	5283	5286	IDT signature evaluated with primary label.
INIT_IDT_SIG_EVAL_TOK	Yes/No	4	5288	5291	IDT signature evaluated with token.
INIT_RESERVED_17	Yes/No	4	5293	5296	Reserved for IBM use.
INIT_RESERVED_18	Yes/No	4	5298	5301	Reserved for IBM use.
INIT_RESERVED_19	Yes/No	4	5303	5306	Reserved for IBM use.
INIT_RESERVED_20	Yes/No	4	5308	5311	Reserved for IBM use.
INIT_RESERVED_21	Yes/No	4	5313	5316	Reserved for IBM use.
INIT_RESERVED_22	Yes/No	4	5318	5321	Reserved for IBM use.

Table 11. Event qualifiers for JOBINIT records		
Event qualifier	Event qualifier number	Event description
SUCCESS	--	Successful initiation (from type 30 record)
TERM	--	Successful termination (from type 30 record)
SUCCESSI	00	Successful initiation.
INVPSWD	01	Not a valid password.
INVGRP	02	Not a valid group.
INVOID	03	Not a valid OIDCARD.
INVTerm	04	Not a valid terminal.
INVAPPL	05	Not a valid application.
REVKUSER	06	User has been revoked.
REVKAUTO	07	User automatically revoked because of excessive password and password phrase attempts.
SUCCEST	08	Successful termination.
UNDFUSER	09	User not defined to RACF.
INSSECL	10	Insufficient security label.
NASECL	11	Not authorized to security label.
RACINITI	12	Successful RACINIT initiation.

Table 11. Event qualifiers for JOBINIT records (continued)

Event qualifier	Event qualifier number	Event description
RACINITD	13	Successful RACINIT deletion.
MOREAUTH	14	User does not have authority to log on while SETROPTS MLQUIET is in effect.
RJENAUTH	15	RJE not authorized.
SURROGTI	16	Surrogate class inactive.
SUBNATHU	17	Submitter not authorized by user.
SUBNATHS	18	Submitter not authorized by security label.
USERNJOB	19	User not authorized to the job.
WINSSECL	20	Warning: Insufficient security label.
WSECLM	21	Warning: security label missing from job.
WNASECL	22	Warning: Not authorized to security label.
SECLNCM	23	Security labels not compatible.
WSECLNCM	24	Warning: security labels not compatible.
PWDEXPR	25	Current password has expired.
INVNPWD	26	Not a valid new password.
EXITFAIL	27	Failed by installation exit.
GRPARVKD	28	Group access revoked.
OIDREQD	29	OIDCARD required.
NJENAUTH	30	NJE job not authorized.
WUKNUPRP	31	Warning: Undefined user from trusted node propagated.
SUCCESSP	32	Successful initiation using a PassTicket.
PTKTREPL	33	Attempted replay of PassTicket.
SECLSRVM	34	Mismatch with server's security label.
REVKINAC	35	User automatically revoked because of inactivity.
INVPHRS	36	Password phrase is not valid.
INVNPHRS	37	New password phrase is not valid.
PHRSEXP	38	Current password phrase has expired.
DIDNOTDF	39	No RACF user ID found for distributed identity.
SUCCESSM	40	Successful IBM Multi-Factor Authentication authentication.
INVMFA	41	Failed IBM Multi-Factor Authentication authentication.
MFAUNAVL	42	Failed authentication because no multi-factor authentication decision could be made for an IBM MFA user who has the NOPWFALLBACK option.
MFAPSUCC	43	IBM MFA partial success: credentials were not incorrect, but a re-authentication is required.
IDTVAlF	44	Identity Token validation error.
IDTF	45	Identity Token build error.
INVIDT	46	Failed Identity Token authentication.

The ACCESS record extension

Table 12 on page 182 describes the format of a record that is created by the access to a resource.

The event qualifiers that can be associated with an access event are shown in Table 13 on page 185.

Table 12. Format of the ACCESS record extension (event code number 02)					
Field name	Type	Length	Position		Comments
			Start	End	
ACC_RES_NAME	Char	255	282	536	Resource name or old resource name.
ACC_REQUEST	Char	8	538	545	Access authority requested.
ACC_GRANT	Char	8	547	554	Access authority granted.
ACC_LEVEL	Integer	3	556	558	Level of the resource.
ACC_VOL	Char	6	560	565	Volume of the resource.
ACC_OLDVOL	Char	6	567	572	OLDVOL of the resource.
ACC_CLASS	Char	8	574	581	Class name.
ACC_APPL	Char	8	583	590	Application name specified.
ACC_TYPE	Char	8	592	599	Type of resource data. Valid values are as follows: "RESOURCE" If the resource name is generic, ACC_TYPE is "RESOURCE". "PROFILE" If the profile name is generic, ACC_TYPE is "PROFILE". " " If both the resource name and profile name are discrete, ACC_TYPE is blank.
ACC_NAME	Char	246	601	846	Resource name or profile name. Note: This field is blank if a discrete profile was used, or when no profile was used, such as when a user accesses their own JES spool files. For discrete profiles, the profile name that was used is the same as the resource name.
ACC_OWN_ID	Char	8	848	855	Name of the profile owner.
ACC_LOGSTR	Char	255	857	1111	LOGSTR= data from the RACROUTE.
ACC_RECVR	Char	8	1113	1120	User ID to whom the data is directed (RECVR= on RACROUTE).
ACC_USER_NAME	Char	20	1122	1141	User name from the ACEE.
ACC_SECL	Char	8	1143	1150	Security label of the resource.
ACC_UTK_ENCR	Yes/No	4	1152	1155	Is the UTOKEN associated with this user encrypted?
ACC_UTK_PRE19	Yes/No	4	1157	1160	Is this a pre-1.9 token?
ACC_UTK_VERPROF	Yes/No	4	1162	1165	Is the VERIFYX propagation flag set?
ACC_UTK_NJEUNUSR	Yes/No	4	1167	1170	Is this the NJE undefined user?
ACC_UTK_LOGUSR	Yes/No	4	1172	1175	Is UAUDIT specified for this user?
ACC_UTK_SPECIAL	Yes/No	4	1177	1180	Is this a SPECIAL user?
ACC_UTK_DEFAULT	Yes/No	4	1182	1185	Is this a default token?
ACC_UTK_UNKNUSR	Yes/No	4	1187	1190	Is this an undefined user?
ACC_UTK_ERROR	Yes/No	4	1192	1195	Is this user token in error?

Table 12. Format of the ACCESS record extension (event code number 02) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
ACC_UTK_TRUSTED	Yes/No	4	1197	1200	Is this user a part of the trusted computing base (TCB)?
ACC_UTK_SESSTYPE	Char	8	1202	1209	The session type of this session.
ACC_UTK_SURROGAT	Yes/No	4	1211	1214	Is this a surrogate user?
ACC_UTK_REMOTE	Yes/No	4	1216	1219	Is this a remote job?
ACC_UTK_PRIV	Yes/No	4	1221	1224	Is this a privileged user ID?
ACC_UTK_SECL	Char	8	1226	1233	The security label of the user.
ACC_UTK_EXECNODE	Char	8	1235	1242	The execution node of the work.
ACC_UTK_SUSER_ID	Char	8	1244	1251	The submitting user ID.
ACC_UTK_SNODE	Char	8	1253	1260	The submitting node.
ACC_UTK_SGRP_ID	Char	8	1262	1269	The submitting group name.
ACC_UTK_SPOE	Char	8	1271	1278	The port of entry.
ACC_UTK_SPCCLASS	Char	8	1280	1287	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ACC_UTK_USER_ID	Char	8	1289	1296	User ID associated with the record.
ACC_UTK_GRP_ID	Char	8	1298	1305	Group name associated with the record.
ACC_UTK_DFT_GRP	Yes/No	4	1307	1310	Is a default group assigned?
ACC_UTK_DFT_SECL	Yes/No	4	1312	1315	Is a default security label assigned?
ACC_RTK_ENCR	Yes/No	4	1317	1320	Is the RTOKEN associated with this user encrypted?
ACC_RTK_PRE19	Yes/No	4	1322	1325	Is this a pre-1.9 token?
ACC_RTK_VERPROF	Yes/No	4	1327	1330	Is the VERIFYX propagation flag set?
ACC_RTK_NJEUNUSR	Yes/No	4	1332	1335	Is this the NJE undefined user?
ACC_RTK_LOGUSR	Yes/No	4	1337	1340	Is UAUDIT specified for this user?
ACC_RTK_SPECIAL	Yes/No	4	1342	1345	Is this a SPECIAL user?
ACC_RTK_DEFAULT	Yes/No	4	1347	1350	Is this a default token?
ACC_RTK_UNKNUSR	Yes/No	4	1352	1355	Is this an undefined user?
ACC_RTK_ERROR	Yes/No	4	1357	1360	Is this user token in error?
ACC_RTK_TRUSTED	Yes/No	4	1362	1365	Is this user a part of the trusted computing base (TCB)?
ACC_RTK_SESSTYPE	Char	8	1367	1374	The session type of this session.
ACC_RTK_SURROGAT	Yes/No	4	1376	1379	Is this a surrogate user?
ACC_RTK_REMOTE	Yes/No	4	1381	1384	Is this a remote job?
ACC_RTK_PRIV	Yes/No	4	1386	1389	Is this a privileged user ID?
ACC_RTK_SECL	Char	8	1391	1398	The security label of the user.
ACC_RTK_EXECNODE	Char	8	1400	1407	The execution node of the work.
ACC_RTK_SUSER_ID	Char	8	1409	1416	The submitting user ID.
ACC_RTK_SNODE	Char	8	1418	1425	The submitting node.
ACC_RTK_SGRP_ID	Char	8	1427	1434	The submitting group name.
ACC_RTK_SPOE	Char	8	1436	1443	The port of entry.

Table 12. Format of the ACCESS record extension (event code number 02) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
ACC_RTK_SPCCLASS	Char	8	1445	1452	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ACC_RTK_USER_ID	Char	8	1454	1461	User ID associated with the record.
ACC_RTK_GRP_ID	Char	8	1463	1470	Group name associated with the record.
ACC_RTK_DFT_GRP	Yes/No	4	1472	1475	Is a default group assigned?
ACC_RTK_DFT_SECL	Yes/No	4	1477	1480	Is a default security label assigned?
ACC_APPC_LINK	Char	16	1482	1497	A key to link together audit record together for a user's APPC transaction processing work.
ACC_DCE_LINK	Char	16	1499	1514	Link to connect DCE records that originate from a single DCE request.
ACC_AUTH_TYPE	Char	13	1516	1528	Defines the type of request. Valid values are "SERVER", "AUTH_CLIENT", "UNAUTH_CLIENT" and "NESTED".
ACC_PDS_DSN	Char	44	1530	1573	Partitioned data set name.
ACC_UTK_NETW	Char	8	1575	1582	The port of entry network name.
ACC_RTK_NETW	Char	8	1584	1591	The network name from the RTOKEN.
ACC_X500_SUBJECT	Char	255	1593	1847	Subject's name associated with this event.
ACC_X500_ISSUER	Char	255	1849	2103	Issuer's name associated with this event.
ACC_USECL	Char	8	2105	2112	Security label of the resource (for DIRAUTH processing only).
ACC_SERV_POENAME	Char	64	2114	2177	SERVAUTH resource or profile name.
ACC_NEST_PRIMARY	Char	8	2179	2186	Primary (client) user ID in nested ACEE.
ACC_CTX_USER	Char	510	2188	2697	Authenticated user name.
ACC_CTX_REG	Char	255	2699	2953	Authenticated user registry name.
ACC_CTX_HOST	Char	128	2955	3082	Authenticated user host name.
ACC_CTX_MECH	Char	16	3084	3099	Authenticated user authentication mechanism object identifier (OID).
ACC_CRITERIA	Char	244	3101	3344	Access criteria.
ACC_IDID_USER	Char	985	3346	4330	Authenticated distributed user name.
ACC_IDID_REG	Char	1021	4332	5352	Authenticated distributed user registry name.
ACC_Reserved_1	Integer	4	5354	5357	
ACC_Reserved_2	Char	8	5359	5366	
ACC_Reserved_3	Char	8	5368	5375	
ACC_LOGSTRX_TYPE	Char	4	5377	5380	Value=0001 is the only value currently supported and indicates that the following triplets contain the identity of the CICS client accessing a resource.
ACC_CICSU_USER_ID	Char	8	5382	5389	CICS client user ID.
ACC_CICSU_X500_SUBJECT	Char	255	5391	5645	CICS client X500 subject if a certificate is provided.
ACC_CICSU_X500_ISSUER	Char	255	5647	5901	CICS client X500 certificate issuer.
ACC_CICSU_IDID_USR_EBC	Char	738	5903	6640	CICS client IDID User in EBCDIC.
ACC_CICSU_IDID_USR_UTF8	Char	246	6642	6887	CICS client IDID user in UTF8.

Table 12. Format of the ACCESS record extension (event code number 02) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
ACC_CICSU_IDID_REG_EBC	Char	765	6889	7653	CICS client IDID registry in EBCDIC.
ACC_CICSU_IDID_REG_UTF8	Char	255	7655	7909	CICS client IDID registry in UTF8.
ACC_CICSU_APPLID	Char	8	7911	7918	CICS client Application ID.
ACC_CICSU_TRANID	Char	4	7920	7923	CICS client Transaction ID.

Table 13. Event qualifiers for access records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful access.
INSAUTH	01	Insufficient authority.
PRFNFD	02	Profile not found; RACFIND specified on macro.
WARNING	03	Access allowed by WARNING.
FPROTALL	04	Failed by PROTECTALL.
WPROTALL	05	PROTECTALL warning.
INSCATG	06	Insufficient category or level.
INSSECL	07	Insufficient security label.
WSECLM	08	Warning: security label missing.
WINSSECL	09	Warning: Insufficient security label.
WNOTCAT	10	Warning: Data set not cataloged, but was required for authority check.
NOTCAT	11	Data set not cataloged.
PRFNFDAI	12	Profile not found.
WINSCATG	13	Warning: Insufficient category or level.
WNONMAIN	14	Warning: Non-MAIN execution environment detected while in ENHANCED PGMSECURITY mode. Conditional access of EXECUTE-controlled program temporarily allowed.
PGMBASIC	15	Conditional access or use of EXECUTE-controlled program allowed through BASIC mode program while in ENHANCED PGMSECURITY mode.

The ADDVOL record extension

Table 14 on page 185 describes the format of a record that is created by the ADDVOL or CHGVOL operations.

The event qualifiers that can be associated with an ADDVOL event are shown in Table 15 on page 187.

Table 14. Format of the ADDVOL record extension (event code 03)					
Field name	Type	Length	Position		Comments
			Start	End	
ADV_RES_NAME	Char	255	282	536	Resource name.
ADV_GRANT	Char	8	538	545	The access authority granted.
ADV_LEVEL	Integer	3	547	549	The level of the resource.
ADV_VOL	Char	6	551	556	Volume of the resource.
ADV_OLDVOL	Char	6	558	563	OLDVOL of the resource.

Table 14. Format of the ADDVOL record extension (event code 03) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
ADV_CLASS	Char	8	565	572	Class name.
ADV_OWN_ID	Char	8	574	581	Name of the profile owner.
ADV_LOGSTR	Char	255	583	837	LOGSTR= data from the RACROUTE
ADV_USER_NAME	Char	20	839	858	User name from the ACEE.
ADV_UTK_ENCR	Yes/No	4	860	863	Is the UTOKEN associated with this user encrypted?
ADV_UTK_PRE19	Yes/No	4	865	868	Is this a pre-1.9 token?
ADV_UTK_VERPROF	Yes/No	4	870	873	Is the VERIFYX propagation flag set?
ADV_UTK_NJEUNUSR	Yes/No	4	875	878	Is this the NJE undefined user?
ADV_UTK_LOGUSR	Yes/No	4	880	883	Is UAUDIT specified for this user?
ADV_UTK_SPECIAL	Yes/No	4	885	888	Is this a SPECIAL user?
ADV_UTK_DEFAULT	Yes/No	4	890	893	Is this a default token?
ADV_UTK_UNKNUSR	Yes/No	4	895	898	Is this an undefined user?
ADV_UTK_ERROR	Yes/No	4	900	903	Is this user token in error?
ADV_UTK_TRUSTED	Yes/No	4	905	908	Is this user a part of the trusted computing base (TCB)?
ADV_UTK_SESSTYPE	Char	8	910	917	The session type of this session.
ADV_UTK_SURROGAT	Yes/No	4	919	922	Is this a surrogate user?
ADV_UTK_REMOTE	Yes/No	4	924	927	Is this a remote job?
ADV_UTK_PRIV	Yes/No	4	929	932	Is this a privileged user ID?
ADV_UTK_SECL	Char	8	934	941	The security label of the user.
ADV_UTK_EXECNODE	Char	8	943	950	The execution node of the work.
ADV_UTK_SUSER_ID	Char	8	952	959	The submitting user ID.
ADV_UTK_SNODE	Char	8	961	968	The submitting node.
ADV_UTK_SGRP_ID	Char	8	970	977	The submitting group name.
ADV_UTK_SPOE	Char	8	979	986	The port of entry.
ADV_UTK_SPCCLASS	Char	8	988	995	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ADV_UTK_USER_ID	Char	8	997	1004	User ID associated with the record.
ADV_UTK_GRP_ID	Char	8	1006	1013	Group name associated with the record.
ADV_UTK_DFT_GRP	Yes/No	4	1015	1018	Is a default group assigned?
ADV_UTK_DFT_SECL	Yes/No	4	1020	1023	Is a default security label assigned?
ADV_APPC_LINK	Char	16	1025	1040	Key to link together APPC records.
ADV_SPECIFIED	Char	1024	1042	2065	The keywords specified.
ADV_UTK_NETW	Char	8	2067	2074	The port of entry network name.
ADV_X500_SUBJECT	Char	255	2076	2330	Subject's name associated with this event.
ADV_X500_ISSUER	Char	255	2332	2586	Issuer's name associated with this event.
ADV_SERV_POENAME	Char	64	2588	2651	SERVAUTH resource or profile name.
ADV_RES_SECL	Char	8	2653	2660	Resource security label.
ADV_CTX_USER	Char	510	2662	3171	Authenticated user name.

Table 14. Format of the ADDVOL record extension (event code 03) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
ADV_CTX_REG	Char	255	3173	3427	Authenticated user registry name.
ADV_CTX_HOST	Char	128	3429	3556	Authenticated user host name.
ADV_CTX_MECH	Char	16	3558	3573	Authenticated user authentication mechanism object identifier (OID).
ADV_IDID_USER	Char	985	3575	4559	Authenticated distributed user name.
ADV_IDID_REG	Char	1021	4561	5581	Authenticated distributed user registry name.

Table 15. Event qualifiers for add volume/change volume records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	The volume was successfully added or changed.
INSAUTH	01	Insufficient authority.
INSSECL	02	Insufficient security label authority.
LESSSPEC	03	A less-specific profile exists with a different security label.

The RENAMEDS record extension

Table 16 on page 187 describes the format of a record that is created by the rename data set, rename SFS file, or rename SFS directory operation.

The event qualifiers that can be associated with a RENAMEDS event are shown in Table 17 on page 188.

Table 16. Format of the RENAMEDS record extension (event code 04)					
Field name	Type	Length	Position		Comments
			Start	End	
REN_RES_NAME	Char	255	282	536	Old resource name.
REN_NEW_RES_NAME	Char	255	538	792	New Resource name.
REN_LEVEL	Integer	3	794	796	The level of the resource.
REN_VOL	Char	6	798	803	Volume of the resource.
REN_CLASS	Char	8	805	812	Class name.
REN_OWN_ID	Char	8	814	821	Name of the profile owner.
REN_LOGSTR	Char	255	823	1077	LOGSTR= data from the RACROUTE
REN_USER_NAME	Char	20	1079	1098	User name from the ACEE.
REN_UTK_ENCR	Yes/No	4	1100	1103	Is the UTKEN associated with this user encrypted?
REN_UTK_PRE19	Yes/No	4	1105	1108	Is this a pre-1.9 token?
REN_UTK_VERPROF	Yes/No	4	1110	1113	Is the VERIFYX propagation flag set?
REN_UTK_NJEUNUSR	Yes/No	4	1115	1118	Is this the NJE undefined user?
REN_UTK_LOGUSR	Yes/No	4	1120	1123	Is UAUDIT specified for this user?
REN_UTK_SPECIAL	Yes/No	4	1125	1128	Is this a SPECIAL user?
REN_UTK_DEFAULT	Yes/No	4	1130	1133	Is this a default token?
REN_UTK_UNKNUSR	Yes/No	4	1135	1138	Is this an undefined user?
REN_UTK_ERROR	Yes/No	4	1140	1143	Is this user token in error?

Table 16. Format of the RENAMEDS record extension (event code 04) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
REN_UTK_TRUSTED	Yes/No	4	1145	1148	Is this user a part of the trusted computing base (TCB)?
REN_UTK_SESSTYPE	Char	8	1150	1157	The session type of this session.
REN_UTK_SURROGAT	Yes/No	4	1159	1162	Is this a surrogate user?
REN_UTK_REMOTE	Yes/No	4	1164	1167	Is this a remote job?
REN_UTK_PRIV	Yes/No	4	1169	1172	Is this a privileged user ID?
REN_UTK_SECL	Char	8	1174	1181	The security label of the user.
REN_UTK_EXECNODE	Char	8	1183	1190	The execution node of the work.
REN_UTK_SUSER_ID	Char	8	1192	1199	The submitting user ID.
REN_UTK_SNODE	Char	8	1201	1208	The submitting node.
REN_UTK_SGRP_ID	Char	8	1210	1217	The submitting group name.
REN_UTK_SPOE	Char	8	1219	1226	The port of entry.
REN_UTK_SPCCLASS	Char	8	1228	1235	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
REN_UTK_USER_ID	Char	8	1237	1244	User ID associated with the record.
REN_UTK_GRP_ID	Char	8	1246	1253	Group name associated with the record.
REN_UTK_DFT_GRP	Yes/No	4	1255	1258	Is a default group assigned?
REN_UTK_DFT_SECL	Yes/No	4	1260	1263	Is a default security label assigned?
REN_APPC_LINK	Char	16	1265	1280	Key to link together APPC records.
REN_SPECIFIED	Char	1024	1282	2305	The keywords specified.
REN_UTK_NETW	Char	8	2307	2314	The port of entry network name.
REN_X500_SUBJECT	Char	255	2316	2570	Subject's name associated with this event.
REN_X500_ISSUER	Char	255	2572	2826	Issuer's name associated with this event.
REN_SERV_POENAME	Char	64	2828	2891	SERVAUTH resource or profile name.
REN_RES_SECL	Char	8	2893	2900	Resource security label.
REN_CTX_USER	Char	510	2902	3411	Authenticated user name.
REN_CTX_REG	Char	255	3413	3667	Authenticated user registry name.
REN_CTX_HOST	Char	128	3669	3796	Authenticated user host name.
REN_CTX_MECH	Char	16	3798	3813	Authenticated user authentication mechanism object identifier (OID).
REN_IDID_USER	Char	985	3815	4799	Authenticated distributed user name.
REN_IDID_REG	Char	1021	4801	5821	Authenticated distributed user registry name.

Table 17. Event qualifiers for RENAMEDS records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful rename.
INVGRP	01	Invalid group.
NOTINGRP	02	User not in group.
INSAUTH	03	Insufficient authority.

Table 17. Event qualifiers for RENAMEDS records (continued)

Event qualifier	Event qualifier number	Event description
ALRDEFD	04	Resource already defined.
NOTRACF	05	User is not RACF-defined.
NOTPROT	06	Resource not protected.
WNOTPROT	07	Warning: Resource not protected
NOT2RACF	08	User in second qualifier is not RACF-defined.
LESSSPEC	09	A less-specific profile exists with a different security label.
INSSECL	10	Insufficient security label authority.
RSNSECL	11	Resource not protected by security label.
NMNSECL	12	New name not protected by security label.
NODOMIN	13	New security label must dominate old security label.
WINSSECL	14	Warning: Insufficient security label authority.
WRSNSECL	15	Warning: Resource not protected by security label.
WNMNSECL	16	Warning: New name not protected by security label.
WNODOMIN	17	Warning: New security label must dominate old security label.

The DELRES record extension

Table 18 on page 189 describes the format of a record that is created by the delete resource operation.

The event qualifiers that may be associated with a DELRES event are shown in Table 19 on page 190.

Table 18. Format of the DELRES record extension (event code 05)

Field name	Type	Length	Position		Comments
			Start	End	
DELR_RES_NAME	Char	255	282	536	Old resource name.
DELR_LEVEL	Integer	3	538	540	The level of the resource.
DELR_VOL	Char	6	542	547	Volume of the resource.
DELR_CLASS	Char	8	549	556	Class name.
DELR_OWN_ID	Char	8	558	565	Name of the profile owner.
DELR_LOGSTR	Char	255	567	821	LOGSTR= data from the RACROUTE
DELR_USER_NAME	Char	20	823	842	User name from the ACEE.
DELR_UTK_ENCR	Yes/No	4	844	847	Is the UTKEN associated with this user encrypted?
DELR_UTK_PRE19	Yes/No	4	849	852	Is this a pre-1.9 token?
DELR_UTK_VERPROF	Yes/No	4	854	857	Is the VERIFYX propagation flag set?
DELR_UTK_NJEUNUSR	Yes/No	4	859	862	Is this the NJE undefined user?
DELR_UTK_LOGUSR	Yes/No	4	864	867	Is UAUDIT specified for this user?
DELR_UTK_SPECIAL	Yes/No	4	869	872	Is this a SPECIAL user?
DELR_UTK_DEFAULT	Yes/No	4	874	877	Is this a default token?
DELR_UTK_UNKNUSR	Yes/No	4	879	882	Is this an undefined user?
DELR_UTK_ERROR	Yes/No	4	884	887	Is this user token in error?

Table 18. Format of the DELRES record extension (event code 05) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
DELR_UTK_TRUSTED	Yes/No	4	889	892	Is this user a part of the trusted computing base (TCB)?
DELR_UTK_SESSTYPE	Char	8	894	901	The session type of this session.
DELR_UTK_SURROGAT	Yes/No	4	903	906	Is this a surrogate user?
DELR_UTK_REMOTE	Yes/No	4	908	911	Is this a remote job?
DELR_UTK_PRIV	Yes/No	4	913	916	Is this a privileged user ID?
DELR_UTK_SECL	Char	8	918	925	The security label of the user.
DELR_UTK_EXECNODE	Char	8	927	934	The execution node of the work.
DELR_UTK_SUSER_ID	Char	8	936	943	The submitting user ID.
DELR_UTK_SNODE	Char	8	945	952	The submitting node.
DELR_UTK_SGRP_ID	Char	8	954	961	The submitting group name.
DELR_UTK_SPOE	Char	8	963	970	The port of entry.
DELR_UTK_SPCLASS	Char	8	972	979	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DELR_UTK_USER_ID	Char	8	981	988	User ID associated with the record.
DELR_UTK_GRP_ID	Char	8	990	997	Group name associated with the record.
DELR_UTK_DFT_GRP	Yes/No	4	999	1002	Is a default group assigned?
DELR_UTK_DFT_SECL	Yes/No	4	1004	1007	Is a default security label assigned?
DELR_APPC_LINK	Char	16	1009	1024	Key to link together APPC records.
DELR_SPECIFIED	Char	1024	1026	2049	Keywords specified.
DELR_UTK_NETW	Char	8	2051	2058	The port of entry network name.
DELR_X500_SUBJECT	Char	255	2060	2314	Subject's name associated with this event.
DELR_X500_ISSUER	Char	255	2316	2570	Issuer's name associated with this event.
DELR_SERV_POENAME	Char	64	2572	2635	SERVAUTH resource or profile name.
DELR_RES_SECL	Char	8	2637	2644	Resource security label.
DELR_CTX_USER	Char	510	2646	3155	Authenticated user name.
DELR_CTX_REG	Char	255	3157	3411	Authenticated user registry name.
DELR_CTX_HOST	Char	128	3413	3540	Authenticated user host name.
DELR_CTX_MECH	Char	16	3542	3557	Authenticated user authentication mechanism object identifier (OID).
DELR_IDID_USER	Char	985	3559	4543	Authenticated distributed user name.
DELR_IDID_REG	Char	1021	4545	5565	Authenticated distributed user registry name.

Table 19. Event qualifiers for delete resource records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	The resource was successfully deleted.
NOTFOUND	01	Resource not found.
INVVOL	02	Invalid volume.

The DELVOL record extension

Table 20 on page 191 describes the format of a record that is created by the delete resource operation.

The event qualifier that can be associated with a DELVOL event is shown in Table 21 on page 192.

Table 20. Format of the DELVOL record extension (event code 06)					
Field name	Type	Length	Position		Comments
			Start	End	
DELV_RES_NAME	Char	255	282	536	Old resource name.
DELV_LEVEL	Integer	3	538	540	The level of the resource.
DELV_VOL	Char	6	542	547	Volume of the resource.
DELV_CLASS	Char	8	549	556	Class name.
DELV_OWN_ID	Char	8	558	565	Name of the profile owner.
DELV_LOGSTR	Char	255	567	821	LOGSTR= data from the RACROUTE
DELV_USER_NAME	Char	20	823	842	User name.
DELV_UTK_ENCR	Yes/No	4	844	847	Is the UTOKEN associated with this user encrypted?
DELV_UTK_PRE19	Yes/No	4	849	852	Is this a pre-1.9 token?
DELV_UTK_VERPROF	Yes/No	4	854	857	Is the VERIFYX propagation flag set?
DELV_UTK_NJEUNUSR	Yes/No	4	859	862	Is this the NJE undefined user?
DELV_UTK_LOGUSR	Yes/No	4	864	867	Is UAUDIT specified for this user?
DELV_UTK_SPECIAL	Yes/No	4	869	872	Is this a SPECIAL user?
DELV_UTK_DEFAULT	Yes/No	4	874	877	Is this a default token?
DELV_UTK_UNKNUSR	Yes/No	4	879	882	Is this an undefined user?
DELV_UTK_ERROR	Yes/No	4	884	887	Is this user token in error?
DELV_UTK_TRUSTED	Yes/No	4	889	892	Is this user a part of the trusted computing base (TCB)?
DELV_UTK_SESSTYPE	Char	8	894	901	The session type of this session.
DELV_UTK_SURROGAT	Yes/No	4	903	906	Is this a surrogate user?
DELV_UTK_REMOTE	Yes/No	4	908	911	Is this a remote job?
DELV_UTK_PRIV	Yes/No	4	913	916	Is this a privileged user ID?
DELV_UTK_SECL	Char	8	918	925	The security label of the user.
DELV_UTK_EXECNODE	Char	8	927	934	The execution node of the work.
DELV_UTK_SUSER_ID	Char	8	936	943	The submitting user ID.
DELV_UTK_SNODE	Char	8	945	952	The submitting node.
DELV_UTK_SGRP_ID	Char	8	954	961	The submitting group name.
DELV_UTK_SPOE	Char	8	963	970	The port of entry.
DELV_UTK_SPCLASS	Char	8	972	979	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DELV_UTK_USER_ID	Char	8	981	988	User ID associated with the record.
DELV_UTK_GRP_ID	Char	8	990	997	Group name associated with the record.
DELV_UTK_DFT_GRP	Yes/No	4	999	1002	Is a default group assigned?
DELV_UTK_DFT_SECL	Yes/No	4	1004	1007	Is a default security label assigned?
DELV_APPC_LINK	Char	16	1009	1024	Key to link together APPC records.
DELV_SPECIFIED	Char	1024	1026	2049	The keywords specified.

Table 20. Format of the DELVOL record extension (event code 06) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
DELV_UTK_NETW	Char	8	2051	2058	The port of entry network name.
DELV_X500_SUBJECT	Char	255	2060	2314	Subject's name associated with this event.
DELV_X500_ISSUER	Char	255	2316	2570	Issuer's name associated with this event.
DELV_SERV_POENAME	Char	64	2572	2635	SERVAUTH resource or profile name.
DELV_RES_SECL	Char	8	2637	2644	Resource security label.
DELV_CTX_USER	Char	510	2646	3155	Authenticated user name.
DELV_CTX_REG	Char	255	3157	3411	Authenticated user registry name.
DELV_CTX_HOST	Char	128	3413	3540	Authenticated user host name.
DELV_CTX_MECH	Char	16	3542	3557	Authenticated user authentication mechanism object identifier (OID).
DELV_IDID_USER	Char	985	3559	4543	Authenticated distributed user name.
DELV_IDID_REG	Char	1021	4545	5565	Authenticated distributed user registry name.

Table 21. Event qualifiers for delete volume records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	The volume was successfully deleted.

The DEFINE record extension

Table 22 on page 192 describes the format of a record that is created by the define resource operation.

The event qualifiers that can be associated with a DEFINE event are shown in Table 23 on page 193.

Table 22. Format of the DEFINE record extension (event code 07)					
Field name	Type	Length	Position		Comments
			Start	End	
DEF_RES_NAME	Char	255	282	536	Old resource name.
DEF_LEVEL	Integer	3	538	540	The level of the resource.
DEF_VOL	Char	6	542	547	Volume of the resource.
DEF_CLASS	Char	8	549	556	Class name.
DEF_MODEL_NAME	Char	255	558	812	Name of the model profile.
DEF_MODEL_VOL	Char	6	814	819	Volser of the model profile.
DEF_OWN_ID	Char	8	821	828	Owner of the profile.
DEF_LOGSTR	Char	255	830	1084	LOGSTR= data from the RACROUTE
DEF_USER_NAME	Char	20	1086	1105	User name.
DEF_UTK_ENCR	Yes/No	4	1107	1110	Is the UTKEN associated with this user encrypted?
DEF_UTK_PRE19	Yes/No	4	1112	1115	Is this a pre-1.9 token?
DEF_UTK_VERPROF	Yes/No	4	1117	1120	Is the VERIFYX propagation flag set?
DEF_UTK_NJEUNUSR	Yes/No	4	1122	1125	Is this the NJE undefined user?
DEF_UTK_LOGUSR	Yes/No	4	1127	1130	Is UAUDIT specified for this user?
DEF_UTK_SPECIAL	Yes/No	4	1132	1135	Is this a SPECIAL user?

Table 22. Format of the DEFINE record extension (event code 07) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
DEF_UTK_DEFAULT	Yes/No	4	1137	1140	Is this a default token?
DEF_UTK_UNKNUSR	Yes/No	4	1142	1145	Is this an undefined user?
DEF_UTK_ERROR	Yes/No	4	1147	1150	Is this user token in error?
DEF_UTK_TRUSTED	Yes/No	4	1152	1155	Is this user a part of the trusted computing base (TCB)?
DEF_UTK_SESSTYPE	Char	8	1157	1164	The session type of this session.
DEF_UTK_SURROGAT	Yes/No	4	1166	1169	Is this a surrogate user?
DEF_UTK_REMOTE	Yes/No	4	1171	1174	Is this a remote job?
DEF_UTK_PRIV	Yes/No	4	1176	1179	Is this a privileged user ID?
DEF_UTK_SECL	Char	8	1181	1188	The security label of the user.
DEF_UTK_EXECNODE	Char	8	1190	1197	The execution node of the work.
DEF_UTK_SUSER_ID	Char	8	1199	1206	The submitting user ID.
DEF_UTK_SNODE	Char	8	1208	1215	The submitting node.
DEF_UTK_SGRP_ID	Char	8	1217	1224	The submitting group name.
DEF_UTK_SPOE	Char	8	1226	1233	The port of entry.
DEF_UTK_SPCCLASS	Char	8	1235	1242	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DEF_UTK_USER_ID	Char	8	1244	1251	User ID associated with the record.
DEF_UTK_GRP_ID	Char	8	1253	1260	Group name associated with the record.
DEF_UTK_DFT_GRP	Yes/No	4	1262	1265	Is a default group assigned?
DEF_UTK_DFT_SECL	Yes/No	4	1267	1270	Is a default security label assigned?
DEF_APPC_LINK	Char	16	1272	1287	Key to link together APPC records.
DEF_SPECIFIED	Char	1024	1289	2312	The keywords specified.
DEF_UTK_NETW	Char	8	2314	2321	The port of entry network name.
DEF_X500_SUBJECT	Char	255	2323	2577	Subject's name associated with this event.
DEF_X500_ISSUER	Char	255	2579	2833	Issuer's name associated with this event.
DEF_SERV_POENAME	Char	64	2835	2898	SERVAUTH resource or profile name.
DEF_RES_SECL	Char	8	2900	2907	Resource security label.
DEF_CTX_USER	Char	510	2909	3418	Authenticated user name.
DEF_CTX_REG	Char	255	3420	3674	Authenticated user registry name.
DEF_CTX_HOST	Char	128	3676	3803	Authenticated user host name.
DEF_CTX_MECH	Char	16	3805	3820	Authenticated user authentication mechanism object identifier (OID).
DEF_IDID_USER	Char	985	3822	4806	Authenticated distributed user name.
DEF_IDID_REG	Char	1021	4808	5828	Authenticated distributed user registry name.

Table 23. Event qualifiers for define resource records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful definition.

Table 23. Event qualifiers for define resource records (continued)		
Event qualifier	Event qualifier number	Event description
UNDGROUP	01	Undefined group.
USNINGRP	02	User not in group.
INSAUTH	03	Insufficient authority.
ALRDEFD	04	Resource already defined.
NOTRACF	05	User is not RACF-defined.
NOTPROT	06	Resource not protected.
WNOTPROT	07	Warning: Resource not protected.
WSECLM	08	Warning: security label missing.
WINSSECL	09	Warning: insufficient security label.
NOT2RACF	10	User in second qualifier is not RACF-defined.
INSSECL	11	Insufficient security label authority.
LESSSPEC	12	A less-specific profile exists with a different security label.

The ADDSD record extension

Table 24 on page 194 describes the format of a record that is created by the ADDSD command.

The event qualifiers that can be associated with an ADDSD command are shown in Table 25 on page 195.

Table 24. Format of the ADDSD record extension (event code 08)					
Field name	Type	Length	Position		Comments
			Start	End	
AD_OWN_ID	Char	8	282	289	Owner of the profile.
AD_USER_NAME	Char	20	291	310	User name.
AD_SECL	Char	8	312	319	The security label associated with the profile.
AD_UTK_ENCR	Yes/No	4	321	324	Is the UTKEN associated with this user encrypted?
AD_UTK_PRE19	Yes/No	4	326	329	Is this a pre-1.9 token?
AD_UTK_VERPROF	Yes/No	4	331	334	Is the VERIFYX propagation flag set?
AD_UTK_NJEUNUSR	Yes/No	4	336	339	Is this the NJE undefined user?
AD_UTK_LOGUSR	Yes/No	4	341	344	Is UAUDIT specified for this user?
AD_UTK_SPECIAL	Yes/No	4	346	349	Is this a SPECIAL user?
AD_UTK_DEFAULT	Yes/No	4	351	354	Is this a default token?
AD_UTK_UNKNUSR	Yes/No	4	356	359	Is this an undefined user?
AD_UTK_ERROR	Yes/No	4	361	364	Is this user token in error?
AD_UTK_TRUSTED	Yes/No	4	366	369	Is this user a part of the trusted computing base (TCB)?
AD_UTK_SESSTYPE	Char	8	371	378	The session type of this session.
AD_UTK_SURROGAT	Yes/No	4	380	383	Is this a surrogate user?
AD_UTK_REMOTE	Yes/No	4	385	388	Is this a remote job?
AD_UTK_PRIV	Yes/No	4	390	393	Is this a privileged user ID?
AD_UTK_SECL	Char	8	395	402	The security label of the user.

Table 24. Format of the ADDSD record extension (event code 08) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
AD_UTK_EXECNODE	Char	8	404	411	The execution node of the work.
AD_UTK_USUSER_ID	Char	8	413	420	The submitting user ID.
AD_UTK_SNODE	Char	8	422	429	The submitting node.
AD_UTK_SGRP_ID	Char	8	431	438	The submitting group name.
AD_UTK_SPOE	Char	8	440	447	The port of entry.
AD_UTK_SPCCLASS	Char	8	449	456	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
AD_UTK_USER_ID	Char	8	458	465	User ID associated with the record.
AD_UTK_GRP_ID	Char	8	467	474	Group name associated with the record.
AD_UTK_DFT_GRP	Yes/No	4	476	479	Is a default group assigned?
AD_UTK_DFT_SECL	Yes/No	4	481	484	Is a default security label assigned?
AD_APPC_LINK	Char	16	486	501	Key to link together APPC records.
AD_SECL_LINK	Char	16	503	518	Key to link together the data sets affected by a change of security label and the command that caused the security label change.
AD_DS_NAME	Char	44	520	563	The data set name.
AD_SPECIFIED	Char	1024	565	1588	The keywords specified.
AD_FAILED	Char	1024	1590	2613	The keywords that failed.
AD_UTK_NETW	Char	8	2615	2622	The port of entry network name.
AD_X500_SUBJECT	Char	255	2624	2878	Subject's name associated with this event.
AD_X500_ISSUER	Char	255	2880	3134	Issuer's name associated with this event.
AD_SERV_POENAME	Char	64	3136	3199	SERVAUTH resource or profile name.
AD_CTX_USER	Char	510	3201	3710	Authenticated user name.
AD_CTX_REG	Char	255	3712	3966	Authenticated user registry name.
AD_CTX_HOST	Char	128	3968	4095	Authenticated user host name.
AD_CTX_MECH	Char	16	4097	4112	Authenticated user authentication mechanism object identifier (OID).
AD_IDID_USER	Char	985	4114	5098	Authenticated distributed user name.
AD_IDID_REG	Char	1021	5100	6120	Authenticated distributed user registry name.

Table 25. Event qualifiers for ADDSD command records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.
SECLSUCC	03	Successful retrieval of data set names.
SECLFAIL	04	Error during retrieval of data set names.

The ADDGROUP record extension

Table 26 on page 196 describes the format of a record that is created by the ADDGROUP command.

The event qualifiers that can be associated with an ADDGROUP command are shown in [Table 27 on page 197](#).

Table 26. Format of the ADDGROUP record extension (event code 09)					
Field name	Type	Length	Position		Comments
			Start	End	
AG_OWN_ID	Char	8	282	289	Owner of the profile.
AG_USER_NAME	Char	20	291	310	User name.
AG_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
AG_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
AG_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
AG_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
AG_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
AG_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
AG_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
AG_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
AG_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
AG_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
AG_UTK_SESTYPE	Char	8	362	369	The session type of this session.
AG_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
AG_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
AG_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
AG_UTK_SECL	Char	8	386	393	The security label of the user.
AG_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
AG_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
AG_UTK_SNODE	Char	8	413	420	The submitting node.
AG_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
AG_UTK_SPOE	Char	8	431	438	The port of entry.
AG_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
AG_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
AG_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
AG_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
AG_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
AG_APPC_LINK	Char	16	477	492	Key to link together APPC records.
AG_GRP_ID	Char	8	494	501	The group name.
AG_SPECIFIED	Char	1024	503	1526	The keywords specified.
AG_FAILED	Char	1024	1528	2551	The keywords that failed.
AG_UTK_NETW	Char	8	2553	2560	The port of entry network name.
AG_X500_SUBJECT	Char	255	2562	2816	Subject's name associated with this event.
AG_X500_ISSUER	Char	255	2818	3072	Issuer's name associated with this event.
AG_SERV_POENAME	Char	64	3074	3137	SERVAUTH resource or profile name.

Table 26. Format of the ADDGROUP record extension (event code 09) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
AG_CTX_USER	Char	510	3139	3648	Authenticated user name.
AG_CTX_REG	Char	255	3650	3904	Authenticated user registry name.
AG_CTX_HOST	Char	128	3906	4033	Authenticated user host name.
AG_CTX_MECH	Char	16	4035	4050	Authenticated user authentication mechanism object identifier (OID).
AG_IDID_USER	Char	985	4052	5036	Authenticated distributed user name.
AG_IDID_REG	Char	1021	5038	6058	Authenticated distributed user registry name.

Table 27. Event qualifiers for ADDGROUP command records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The ADDUSER record extension

Table 28 on page 197 describes the format of a record that is created by the ADDUSER command.

The event qualifiers that can be associated with an ADDUSER command are shown in Table 29 on page 198.

Table 28. Format of the ADDUSER record extension (event code 10)					
Field name	Type	Length	Position		Comments
			Start	End	
AU_OWN_ID	Char	8	282	289	Owner of the profile.
AU_USER_NAME	Char	20	291	310	User name.
AU_UTK_ENCR	Yes/No	4	312	315	Is the UTKEN associated with this user encrypted?
AU_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
AU_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
AU_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
AU_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
AU_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
AU_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
AU_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
AU_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
AU_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
AU_UTK_SESTYPE	Char	8	362	369	The session type of this session.
AU_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
AU_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
AU_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
AU_UTK_SECL	Char	8	386	393	The security label of the user.

Table 28. Format of the ADDUSER record extension (event code 10) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
AU_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
AU_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
AU_UTK_SNODE	Char	8	413	420	The submitting node.
AU_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
AU_UTK_SPOE	Char	8	431	438	The port of entry.
AU_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
AU_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
AU_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
AU_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
AU_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
AU_APPC_LINK	Char	16	477	492	Key to link together APPC records.
AU_NOAUTH_CLAUTH	Yes/No	4	494	497	Were violations detected because the user issuing the command lacked the CLAUTH authority in the user class?
AU_NOAUTH_GROUP	Yes/No	4	499	502	Were violations detected because the user issuing the command lacked the authority within the group?
AU_USER_ID	Char	8	504	511	The user ID.
AU_SPECIFIED	Char	1024	513	1536	The keywords specified.
AU_FAILED	Char	1024	1538	2561	The keywords that failed.
AU_IGNORED	Char	1024	2563	3586	The keywords ignored.
AU_UTK_NETW	Char	8	3588	3595	The port of entry network name.
AU_X500_SUBJECT	Char	255	3597	3851	Subject's name associated with this event.
AU_X500_ISSUER	Char	255	3853	4107	Issuer's name associated with this event.
AU_SERV_POENAME	Char	64	4109	4172	SERVAUTH resource or profile name.
AU_CTX_USER	Char	510	4174	4683	Authenticated user name.
AU_CTX_REG	Char	255	4685	4939	Authenticated user registry name.
AU_CTX_HOST	Char	128	4941	5068	Authenticated user host name.
AU_CTX_MECH	Char	16	5070	5085	Authenticated user authentication mechanism object identifier (OID).
AU_IDID_USER	Char	985	5087	6071	Authenticated distributed user name.
AU_IDID_REG	Char	1021	6073	7093	Authenticated distributed user registry name.

Table 29. Event qualifiers for ADDUSER command records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The ALTDSD record extension

Table 30 on page 199 describes the format of a record that is created by the ALTDSD command.

The event qualifiers that can be associated with an ALTDSD command are shown in Table 31 on page 200.

Table 30. Format of the ALTDSD record extension (event code 11)					
Field name	Type	Length	Position		Comments
			Start	End	
ALD_OWN_ID	Char	8	282	289	Owner of the profile.
ALD_USER_NAME	Char	20	291	310	User name.
ALD_OLD_SECL	Char	8	312	319	The security label that is being deleted from the profile.
ALD_UTK_ENCR	Yes/No	4	321	324	Is the UTOKEN associated with this user encrypted?
ALD_UTK_PRE19	Yes/No	4	326	329	Is this a pre-1.9 token?
ALD_UTK_VERPROF	Yes/No	4	331	334	Is the VERIFYX propagation flag set?
ALD_UTK_NJEUNUSR	Yes/No	4	336	339	Is this the NJE undefined user?
ALD_UTK_LOGUSR	Yes/No	4	341	344	Is UAUDIT specified for this user?
ALD_UTK_SPECIAL	Yes/No	4	346	349	Is this a SPECIAL user?
ALD_UTK_DEFAULT	Yes/No	4	351	354	Is this a default token?
ALD_UTK_UNKNUSR	Yes/No	4	356	359	Is this an undefined user?
ALD_UTK_ERROR	Yes/No	4	361	364	Is this user token in error?
ALD_UTK_TRUSTED	Yes/No	4	366	369	Is this user a part of the trusted computing base (TCB)?
ALD_UTK_SESSTYPE	Char	8	371	378	The session type of this session.
ALD_UTK_SURROGAT	Yes/No	4	380	383	Is this a surrogate user?
ALD_UTK_REMOTE	Yes/No	4	385	388	Is this a remote job?
ALD_UTK_PRIV	Yes/No	4	390	393	Is this a privileged user ID?
ALD_UTK_SECL	Char	8	395	402	The security label of the user.
ALD_UTK_EXECCODE	Char	8	404	411	The execution node of the work.
ALD_UTK_SUSER_ID	Char	8	413	420	The submitting user ID.
ALD_UTK_SNODE	Char	8	422	429	The submitting node.
ALD_UTK_SGRP_ID	Char	8	431	438	The submitting group name.
ALD_UTK_SPOE	Char	8	440	447	The port of entry.
ALD_UTK_SPCCLASS	Char	8	449	456	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ALD_UTK_USER_ID	Char	8	458	465	User ID associated with the record.
ALD_UTK_GRP_ID	Char	8	467	474	Group name associated with the record.
ALD_UTK_DFT_GRP	Yes/No	4	476	479	Is a default group assigned?
ALD_UTK_DFT_SECL	Yes/No	4	481	484	Is a default security label assigned?
ALD_APPC_LINK	Char	16	486	501	Key to link together APPC records.
ALD_SECL_LINK	Char	16	503	518	Key to link together the data sets affected by a change of security label and the command that caused the security label change.
ALD_DS_NAME	Char	44	520	563	The data set name.
ALD_SPECIFIED	Char	1024	565	1588	The keywords specified.

Table 30. Format of the ALTDSD record extension (event code 11) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
ALD_FAILED	Char	1024	1590	2613	The keywords that failed.
ALD_IGNORED	Char	1024	2615	3638	The keywords ignored.
ALD_UTK_NETW	Char	8	3640	3647	The port of entry network name.
ALD_X500_SUBJECT	Char	255	3649	3903	Subject's name associated with this event.
ALD_X500_ISSUER	Char	255	3905	4159	Issuer's name associated with this event.
ALD_SERV_POENAME	Char	64	4161	4224	SERVAUTH resource or profile name.
ALD_CTX_USER	Char	510	4226	4735	Authenticated user name.
ALD_CTX_REG	Char	255	4737	4991	Authenticated user registry name.
ALD_CTX_HOST	Char	128	4993	5120	Authenticated user host name.
ALD_CTX_MECH	Char	16	5122	5137	Authenticated user authentication mechanism object identifier (OID).
ALD_IDID_USER	Char	985	5139	6123	Authenticated distributed user name.
ALD_IDID_REG	Char	1021	6125	7145	Authenticated distributed user registry name.

Table 31. Event qualifiers for ADDSD command records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.
SECLSUCC	03	Successful retrieval of data set names.
SECLFAIL	04	Error during retrieval of data set names.

The ALTGROUP record extension

Table 32 on page 200 describes the format of a record that is created by the ALTGROUP command.

The event qualifiers that can be associated with an ALTGROUP command are shown in Table 33 on page 202.

Table 32. Format of the ALTGROUP record extension (event code 12)					
Field name	Type	Length	Position		Comments
			Start	End	
ALG_OWN_ID	Char	8	282	289	Owner of the profile.
ALG_USER_NAME	Char	20	291	310	User name.
ALG_UTK_ENCR	Yes/No	4	312	315	Is the UTKEN associated with this user encrypted?
ALG_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
ALG_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
ALG_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
ALG_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
ALG_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
ALG_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?

Table 32. Format of the ALTGROUP record extension (event code 12) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
ALG_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
ALG_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
ALG_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
ALG_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
ALG_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
ALG_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
ALG_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
ALG_UTK_SECL	Char	8	386	393	The security label of the user.
ALG_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
ALG_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
ALG_UTK_SNODE	Char	8	413	420	The submitting node.
ALG_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
ALG_UTK_SPOE	Char	8	431	438	The port of entry.
ALG_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ALG_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
ALG_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
ALG_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
ALG_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
ALG_APPC_LINK	Char	16	477	492	Key to link together APPC records.
ALG_GRP_ID	Char	8	494	501	The group name.
ALG_SPECIFIED	Char	1024	503	1526	The keywords specified.
ALG_FAILED	Char	1024	1528	2551	The keywords that failed.
ALG_IGNORED	Char	1024	2553	3576	The keywords ignored.
ALG_UTK_NETW	Char	8	3578	3585	The port of entry network name.
ALG_X500_SUBJECT	Char	255	3587	3841	Subject's name associated with this event.
ALG_X500_ISSUER	Char	255	3843	4097	Issuer's name associated with this event.
ALG_SERV_POENAME	Char	64	4099	4162	SERVAUTH resource or profile name.
ALG_CTX_USER	Char	510	4164	4673	Authenticated user name.
ALG_CTX_REG	Char	255	4675	4929	Authenticated user registry name.
ALG_CTX_HOST	Char	128	4931	5058	Authenticated user host name.
ALG_CTX_MECH	Char	16	5060	5075	Authenticated user authentication mechanism object identifier (OID).
ALG_IDID_USER	Char	985	5077	6061	Authenticated distributed user name.
ALG_IDID_REG	Char	1021	6063	7083	Authenticated distributed user registry name.

Table 33. Event qualifiers for ALTGROUP command records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The ALTUSER record extension

Table 34 on page 202 describes the format of a record that is created by the ALTUSER command.

The event qualifiers that can be associated with an ALTUSER command are shown in Table 35 on page 203.

Table 34. Format of the ALTUSER record extension (event code 13)					
Field name	Type	Length	Position		Comments
			Start	End	
ALU_OWN_ID	Char	8	282	289	Owner of the profile.
ALU_USER_NAME	Char	20	291	310	User name.
ALU_OLD_SECL	Char	8	312	319	The security label that is being deleted from the profile.
ALU_UTK_ENCR	Yes/No	4	321	324	Is the UTKEN associated with this user encrypted?
ALU_UTK_PRE19	Yes/No	4	326	329	Is this a pre-1.9 token?
ALU_UTK_VERPROF	Yes/No	4	331	334	Is the VERIFYX propagation flag set?
ALU_UTK_NJEUNUSR	Yes/No	4	336	339	Is this the NJE undefined user?
ALU_UTK_LOGUSR	Yes/No	4	341	344	Is UAUDIT specified for this user?
ALU_UTK_SPECIAL	Yes/No	4	346	349	Is this a SPECIAL user?
ALU_UTK_DEFAULT	Yes/No	4	351	354	Is this a default token?
ALU_UTK_UNKNUSR	Yes/No	4	356	359	Is this an undefined user?
ALU_UTK_ERROR	Yes/No	4	361	364	Is this user token in error?
ALU_UTK_TRUSTED	Yes/No	4	366	369	Is this user a part of the trusted computing base (TCB)?
ALU_UTK_SESSTYPE	Char	8	371	378	The session type of this session.
ALU_UTK_SURROGAT	Yes/No	4	380	383	Is this a surrogate user?
ALU_UTK_REMOTE	Yes/No	4	385	388	Is this a remote job?
ALU_UTK_PRIV	Yes/No	4	390	393	Is this a privileged user ID?
ALU_UTK_SECL	Char	8	395	402	The security label of the user.
ALU_UTK_EXECNODE	Char	8	404	411	The execution node of the work.
ALU_UTK_SUSER_ID	Char	8	413	420	The submitting user ID.
ALU_UTK_SNODE	Char	8	422	429	The submitting node.
ALU_UTK_SGRP_ID	Char	8	431	438	The submitting group name.
ALU_UTK_SPOE	Char	8	440	447	The port of entry.
ALU_UTK_SPCCLASS	Char	8	449	456	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ALU_UTK_USER_ID	Char	8	458	465	User ID associated with the record.
ALU_UTK_GRP_ID	Char	8	467	474	Group name associated with the record.

Table 34. Format of the ALTUSER record extension (event code 13) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
ALU_UTK_DFT_GRP	Yes/No	4	476	479	Is a default group assigned?
ALU_UTK_DFT_SECL	Yes/No	4	481	484	Is a default security label assigned?
ALU_APPC_LINK	Char	16	486	501	Key to link together APPC records.
ALU_NOAUTH_CLAUTH	Yes/No	4	503	506	Were violations detected because the user issuing the command lacked the CLAUTH authority in the user class?
ALU_NOAUTH_GROUP	Yes/No	4	508	511	Were violations detected because the user issuing the command lacked the authority within the group?
ALU_NOAUTH_PROF	Yes/No	4	513	516	Were violations detected because the user issuing the command lacked authority to the profile?
ALU_USER_ID	Char	8	518	525	The user ID.
ALU_SPECIFIED	Char	1024	527	1550	The keywords specified.
ALU_FAILED	Char	1024	1552	2575	The keywords that failed.
ALU_IGNORED	Char	1024	2577	3600	The keywords ignored.
ALU_UTK_NETW	Char	8	3602	3609	The port of entry network name.
ALU_X500_SUBJECT	Char	255	3611	3865	Subject's name associated with this event.
ALU_X500_ISSUER	Char	255	3867	4121	Issuer's name associated with this event.
ALU_SERV_POENAME	Char	64	4123	4186	SERVAUTH resource or profile name.
ALU_CTX_USER	Char	510	4188	4697	Authenticated user name.
ALU_CTX_REG	Char	255	4699	4953	Authenticated user registry name.
ALU_CTX_HOST	Char	128	4955	5082	Authenticated user host name.
ALU_CTX_MECH	Char	16	5084	5099	Authenticated user authentication mechanism object identifier (OID).
ALU_IDID_USER	Char	985	5101	6085	Authenticated distributed user name.
ALU_IDID_REG	Char	1021	6087	7101	Authenticated distributed user registry name.

Table 35. Event qualifiers for ALTUSER command records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The CONNECT record extension

Table 36 on page 203 describes the format of a record that is created by the CONNECT command.

The event qualifiers that can be associated with a CONNECT command are shown in Table 37 on page 205.

Table 36. Format of the CONNECT record extension (event code 14)					
Field name	Type	Length	Position		Comments
			Start	End	
CON_OWN_ID	Char	8	282	289	Owner of the profile.

Table 36. Format of the CONNECT record extension (event code 14) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
CON_USER_NAME	Char	20	291	310	User name.
CON_UTK_ENCR	Yes/No	4	312	315	Is the UTKEN associated with this user encrypted?
CON_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
CON_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
CON_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
CON_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
CON_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
CON_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
CON_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
CON_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
CON_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
CON_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
CON_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
CON_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
CON_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
CON_UTK_SECL	Char	8	386	393	The security label of the user.
CON_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
CON_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
CON_UTK_SNODE	Char	8	413	420	The submitting node.
CON_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
CON_UTK_SPOE	Char	8	431	438	The port of entry.
CON_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
CON_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
CON_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
CON_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
CON_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
CON_APPC_LINK	Char	16	477	492	Key to link together APPC records.
CON_USER_ID	Char	8	494	501	The user ID that is being connected.
CON_SPECIFIED	Char	1024	503	1526	The keywords specified.
CON_FAILED	Char	1024	1528	2551	The keywords ignored.
CON_UTK_NETW	Char	8	2553	2560	The port of entry network name.
CON_X500_SUBJECT	Char	255	2562	2816	Subject's name associated with this event.
CON_X500_ISSUER	Char	255	2818	3072	Issuer's name associated with this event.
CON_SERV_POENAME	Char	64	3074	3137	SERVAUTH resource or profile name.
CON_CTX_USER	Char	510	3139	3648	Authenticated user name.
CON_CTX_REG	Char	255	3650	3904	Authenticated user registry name.
CON_CTX_HOST	Char	128	3906	4033	Authenticated user host name.

Table 36. Format of the CONNECT record extension (event code 14) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
CON_CTX_MECH	Char	16	4035	4050	Authenticated user authentication mechanism object identifier (OID).
CON_IDID_USER	Char	985	4052	5036	Authenticated distributed user name.
CON_IDID_REG	Char	1021	5038	6058	Authenticated distributed user registry name.

Table 37. Event qualifiers for CONNECT command records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The DELDSD record extension

Table 38 on page 205 describes the format of a record that is created by the DELDSD command.

The event qualifiers that can be associated with a DELDSD command are shown in Table 39 on page 206.

Table 38. Format of the DELDSD record extension (event code 15)					
Field name	Type	Length	Position		Comments
			Start	End	
DELD_OWN_ID	Char	8	282	289	Owner of the profile.
DELD_USER_NAME	Char	20	291	310	User name.
DELD_OLD_SECL	Char	8	312	319	The security label that is being deleted.
DELD_UTK_ENCR	Yes/No	4	321	324	Is the UTOKEN associated with this user encrypted?
DELD_UTK_PRE19	Yes/No	4	326	329	Is this a pre-1.9 token?
DELD_UTK_VERPROF	Yes/No	4	331	334	Is the VERIFYX propagation flag set?
DELD_UTK_NJEUNUSR	Yes/No	4	336	339	Is this the NJE undefined user?
DELD_UTK_LOGUSR	Yes/No	4	341	344	Is UAUDIT specified for this user?
DELD_UTK_SPECIAL	Yes/No	4	346	349	Is this a SPECIAL user?
DELD_UTK_DEFAULT	Yes/No	4	351	354	Is this a default token?
DELD_UTK_UNKNUSR	Yes/No	4	356	359	Is this an undefined user?
DELD_UTK_ERROR	Yes/No	4	361	364	Is this user token in error?
DELD_UTK_TRUSTED	Yes/No	4	366	369	Is this user a part of the trusted computing base (TCB)?
DELD_UTK_SESSTYPE	Char	8	371	378	The session type of this session.
DELD_UTK_SURROGAT	Yes/No	4	380	383	Is this a surrogate user?
DELD_UTK_REMOTE	Yes/No	4	385	388	Is this a remote job?
DELD_UTK_PRIV	Yes/No	4	390	393	Is this a privileged user ID?
DELD_UTK_SECL	Char	8	395	402	The security label of the user.
DELD_UTK_EXECNODE	Char	8	404	411	The execution node of the work.
DELD_UTK_SUSER_ID	Char	8	413	420	The submitting user ID.

Table 38. Format of the DELDSD record extension (event code 15) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
DELD_UTK_SNODE	Char	8	422	429	The submitting node.
DELD_UTK_SGRP_ID	Char	8	431	438	The submitting group name.
DELD_UTK_SPOE	Char	8	440	447	The port of entry.
DELD_UTK_SPCCLASS	Char	8	449	456	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DELD_UTK_USER_ID	Char	8	458	465	User ID associated with the record.
DELD_UTK_GRP_ID	Char	8	467	474	Group name associated with the record.
DELD_UTK_DFT_GRP	Yes/No	4	476	479	Is a default group assigned?
DELD_UTK_DFT_SECL	Yes/No	4	481	484	Is a default security label assigned
DELD_APPC_LINK	Char	16	486	501	Key to link together APPC records.
DELD_SECL_LINK	Char	16	503	518	Key to link together the data sets affected by a change of security label and the command that caused the security label change.
DELD_DS_NAME	Char	44	520	563	The data set profile that is being deleted.
DELD_SPECIFIED	Char	1024	565	1588	The keywords specified.
DELD_FAILED	Char	1024	1590	2613	The keywords that failed.
DELD_UTK_NETW	Char	8	2615	2622	The port of entry network name.
DELD_X500_SUBJECT	Char	255	2624	2878	Subject's name associated with this event.
DELD_X500_ISSUER	Char	255	2880	3134	Issuer's name associated with this event.
DELD_SERV_POENAME	Char	64	3136	3199	SERVAUTH resource or profile name.
DELD_CTX_USER	Char	510	3201	3710	Authenticated user name.
DELD_CTX_REG	Char	255	3712	3966	Authenticated user registry name.
DELD_CTX_HOST	Char	128	3968	4095	Authenticated user host name.
DELD_CTX_MECH	Char	16	4097	4112	Authenticated user authentication mechanism object identifier (OID).
DELD_IDID_USER	Char	985	4114	5098	Authenticated distributed user name.
DELD_IDID_REG	Char	1021	5100	6120	Authenticated distributed user registry name.

Table 39. Event qualifiers for DELDSD command records

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

The DELGROUP record extension

Table 40 on page 207 describes the format of a record that is created by the DELGROUP command.

The event qualifiers that can be associated with a DELGROUP command are shown in Table 41 on page 208.

Table 40. Format of the DELGROUP record extension (event code 16)					
Field name	Type	Length	Position		Comments
			Start	End	
DELG_OWN_ID	Char	8	282	289	Owner of the profile.
DELG_USER_NAME	Char	20	291	310	User name.
DELG_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
DELG_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
DELG_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
DELG_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
DELG_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
DELG_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
DELG_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
DELG_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
DELG_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
DELG_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
DELG_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
DELG_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
DELG_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
DELG_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
DELG_UTK_SECL	Char	8	386	393	The security label of the user.
DELG_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
DELG_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
DELG_UTK_SNODE	Char	8	413	420	The submitting node.
DELG_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
DELG_UTK_SPOE	Char	8	431	438	The port of entry.
DELG_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DELG_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
DELG_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
DELG_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
DELG_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
DELG_APPC_LINK	Char	16	477	492	Key to link together APPC records.
DELG_GRP_ID	Char	8	494	501	The group that is being deleted.
DELG_SPECIFIED	Char	1024	503	1526	The keywords specified.
DELG_UTK_NETW	Char	8	1528	1535	The port of entry network name.
DELG_X500_SUBJECT	Char	255	1537	1791	Subject's name associated with this event.
DELG_X500_ISSUER	Char	255	1793	2047	Issuer's name associated with this event.
DELG_SERV_POENAME	Char	64	2049	2112	SERVAUTH resource or profile name.
DELG_CTX_USER	Char	510	2114	2623	Authenticated user name.
DELG_CTX_REG	Char	255	2625	2879	Authenticated user registry name.
DELG_CTX_HOST	Char	128	2881	3008	Authenticated user host name.

Table 40. Format of the DELGROUP record extension (event code 16) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
DELG_CTX_MECH	Char	16	3010	3025	Authenticated user authentication mechanism object identifier (OID).
DELG_IDID_USER	Char	985	3027	4011	Authenticated distributed user name.
DELG_IDID_REG	Char	1021	4013	5033	Authenticated distributed user registry name.

Table 41. Event qualifiers for DELGROUP commands records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The DELUSER record extension

Table 42 on page 208 describes the format of a record that is created by the DELUSER command.

The event qualifiers that can be associated with a DELUSER command are shown in Table 43 on page 209.

Table 42. Format of the DELUSER record extension (event code 17)					
Field name	Type	Length	Position		Comments
			Start	End	
DELU_OWN_ID	Char	8	282	289	Owner of the profile.
DELU_USER_NAME	Char	20	291	310	User name.
DELU_UTK_ENCR	Yes/No	4	312	315	Is the UTKEN associated with this user encrypted?
DELU_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
DELU_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
DELU_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
DELU_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
DELU_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
DELU_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
DELU_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
DELU_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
DELU_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
DELU_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
DELU_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
DELU_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
DELU_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
DELU_UTK_SECL	Char	8	386	393	The security label of the user.
DELU_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
DELU_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
DELU_UTK_SNODE	Char	8	413	420	The submitting node.

Table 42. Format of the DELUSER record extension (event code 17) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
DELU_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
DELU_UTK_SPOE	Char	8	431	438	The port of entry.
DELU_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DELU_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
DELU_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
DELU_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
DELU_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
DELU_APPC_LINK	Char	16	477	492	Key to link together APPC records.
DELU_USER_ID	Char	8	494	501	The user ID that is being deleted.
DELU_SPECIFIED	Char	1024	503	1526	The keywords specified.
DELU_UTK_NETW	Char	8	1528	1535	The port of entry network name.
DELU_X500_SUBJECT	Char	255	1537	1791	Subject's name associated with this event.
DELU_X500_ISSUER	Char	255	1793	2047	Issuer's name associated with this event.
DELU_SERV_POENAME	Char	64	2049	2112	SERVAUTH resource or profile name.
DELU_CTX_USER	Char	510	2114	2623	Authenticated user name.
DELU_CTX_REG	Char	255	2625	2879	Authenticated user registry name.
DELU_CTX_HOST	Char	128	2881	3008	Authenticated user host name.
DELU_CTX_MECH	Char	16	3010	3025	Authenticated user authentication mechanism object identifier (OID).
DELU_IDID_USER	Char	985	3027	4011	Authenticated distributed user name.
DELU_IDID_REG	Char	1021	4013	5033	Authenticated distributed user registry name.

Table 43. Event qualifiers for DELUSER command records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The PASSWORD record extension

Table 44 on page 209 describes the format of a record that is created by the PASSWORD command.

The event qualifiers that can be associated with a PASSWORD command are shown in Table 45 on page 211.

Table 44. Format of the PASSWORD record extension (event code 18)					
Field name	Type	Length	Position		Comments
			Start	End	
PWD_OWN_ID	Char	8	282	289	Owner of the profile.
PWD_USER_NAME	Char	20	291	310	User name.
PWD_UTK_ENCR	Yes/No	4	312	315	Is the UTKEN associated with this user encrypted?

Table 44. Format of the PASSWORD record extension (event code 18) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
PWD_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
PWD_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
PWD_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
PWD_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
PWD_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
PWD_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
PWD_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
PWD_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
PWD_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
PWD_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
PWD_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
PWD_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
PWD_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
PWD_UTK_SECL	Char	8	386	393	The security label of the user.
PWD_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
PWD_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
PWD_UTK_SNODE	Char	8	413	420	The submitting node.
PWD_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
PWD_UTK_SPOE	Char	8	431	438	The port of entry.
PWD_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
PWD_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
PWD_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
PWD_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
PWD_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
PWD_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
PWD_SPECIFIED	Char	1024	494	1517	The keywords specified.
PWD_FAILED	Char	1024	1519	2542	The keywords that failed.
PWD_IGNORED	Char	1024	2544	3567	The keywords ignored.
PWD_UTK_NETW	Char	8	3569	3576	The port of entry network name.
PWD_X500_SUBJECT	Char	255	3578	3832	Subject's name associated with this event.
PWD_X500_ISSUER	Char	255	3834	4088	Issuer's name associated with this event.
PWD_SERV_POENAME	Char	64	4090	4153	SERVAUTH resource or profile name.
PWD_CTX_USER	Char	510	4155	4664	Authenticated user name.
PWD_CTX_REG	Char	255	4666	4920	Authenticated user registry name.
PWD_CTX_HOST	Char	128	4922	5049	Authenticated user host name.
PWD_CTX_MECH	Char	16	5051	5066	Authenticated user authentication mechanism object identifier (OID).

Table 44. Format of the PASSWORD record extension (event code 18) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
PWD_IDID_USER	Char	985	5068	6052	Authenticated distributed user name.
PWD_IDID_REG	Char	1021	6054	7074	Authenticated distributed user registry name.

Table 45. Event qualifiers for PASSWORD command records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The PERMIT record extension

Table 46 on page 211 describes the format of a record that is created by the PERMIT command.

The event qualifiers that can be associated with a PERMIT command are shown in Table 47 on page 212.

Table 46. Format of the PERMIT record extension (event code 19)					
Field name	Type	Length	Position		Comments
			Start	End	
PERM_CLASS	Char	8	282	289	Class name.
PERM_OWN_ID	Char	8	291	298	Owner of the profile.
PERM_USER_NAME	Char	20	300	319	User name.
PERM_UTK_ENCR	Yes/No	4	321	324	Is the UTOKEN associated with this user encrypted?
PERM_UTK_PRE19	Yes/No	4	326	329	Is this a pre-1.9 token?
PERM_UTK_VERPROF	Yes/No	4	331	334	Is the VERIFYX propagation flag set?
PERM_UTK_NJEUNUSR	Yes/No	4	336	339	Is this the NJE undefined user?
PERM_UTK_LOGUSR	Yes/No	4	341	344	Is UAUDIT specified for this user?
PERM_UTK_SPECIAL	Yes/No	4	346	349	Is this a SPECIAL user?
PERM_UTK_DEFAULT	Yes/No	4	351	354	Is this a default token?
PERM_UTK_UNKNUSR	Yes/No	4	356	359	Is this an undefined user?
PERM_UTK_ERROR	Yes/No	4	361	364	Is this user token in error?
PERM_UTK_TRUSTED	Yes/No	4	366	369	Is this user a part of the trusted computing base (TCB)?
PERM_UTK_SESTYPE	Char	8	371	378	The session type of this session.
PERM_UTK_SURROGAT	Yes/No	4	380	383	Is this a surrogate user?
PERM_UTK_REMOTE	Yes/No	4	385	388	Is this a remote job?
PERM_UTK_PRIV	Yes/No	4	390	393	Is this a privileged user ID?
PERM_UTK_SECL	Char	8	395	402	The security label of the user.
PERM_UTK_EXECNODE	Char	8	404	411	The execution node of the work.
PERM_UTK_SUSER_ID	Char	8	413	420	The submitting user ID.
PERM_UTK_SNODE	Char	8	422	429	The submitting node.
PERM_UTK_SGRP_ID	Char	8	431	438	The submitting group name.

Table 46. Format of the PERMIT record extension (event code 19) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
PERM_UTK_SPOE	Char	8	440	447	The port of entry.
PERM_UTK_SPCLASS	Char	8	449	456	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
PERM_UTK_USER_ID	Char	8	458	465	User ID associated with the record.
PERM_UTK_GRP_ID	Char	8	467	474	Group name associated with the record.
PERM_UTK_DFT_GRP	Yes/No	4	476	479	Is a default group assigned?
PERM_UTK_DFT_SECL	Yes/No	4	481	484	Is a default security label assigned?
PERM_APPC_LINK	Char	16	486	501	Key to link together APPC records.
PERM_RES_NAME	Char	255	503	757	The resource name
PERM_SPECIFIED	Char	1024	759	1782	The keywords specified.
PERM_FAILED	Char	1024	1784	2807	The keywords that failed.
PERM_IGNORED	Char	1024	2809	3832	The keywords ignored.
PERM_UTK_NETW	Char	8	3834	3841	The port of entry network name.
PERM_X500_SUBJECT	Char	255	3843	4097	Subject's name associated with this event.
PERM_X500_ISSUER	Char	255	4099	4353	Issuer's name associated with this event.
PERM_SERV_POENAME	Char	64	4355	4418	SERVAUTH resource or profile name.
PERM_CTX_USER	Char	510	4420	4929	Authenticated user name.
PERM_CTX_REG	Char	255	4931	5185	Authenticated user registry name.
PERM_CTX_HOST	Char	128	5187	5314	Authenticated user host name.
PERM_CTX_MECH	Char	16	5316	5331	Authenticated user authentication mechanism object identifier (OID).
PERM_IDID_USER	Char	985	5333	6317	Authenticated distributed user name.
PERM_IDID_REG	Char	1021	6319	7339	Authenticated distributed user registry name.

Table 47. Event qualifiers for PERMIT command records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The RALTER record extension

Table 48 on page 212 describes the format of a record that is created by the RALTER command.

The event qualifiers that can be associated with a RALTER command are shown in Table 49 on page 214.

Table 48. Format of the RALTER record extension (event code 20)					
Field name	Type	Length	Position		Comments
			Start	End	
RALT_CLASS	Char	8	282	289	Class name.
RALT_OWN_ID	Char	8	291	298	Owner of the profile.

Table 48. Format of the RALTER record extension (event code 20) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RALT_USER_NAME	Char	20	300	319	User name.
RALT_OLD_SECL	Char	8	321	328	The security label being deleted from the file.
RALT_UTK_ENCR	Yes/No	4	330	333	Is the UTOKEN associated with this user encrypted?
RALT_UTK_PRE19	Yes/No	4	335	338	Is this a pre-1.9 token?
RALT_UTK_VERPROF	Yes/No	4	340	343	Is the VERIFYX propagation flag set?
RALT_UTK_NJEUNUSR	Yes/No	4	345	348	Is this the NJE undefined user?
RALT_UTK_LOGUSR	Yes/No	4	350	353	Is UAUDIT specified for this user?
RALT_UTK_SPECIAL	Yes/No	4	355	358	Is this a SPECIAL user?
RALT_UTK_DEFAULT	Yes/No	4	360	363	Is this a default token?
RALT_UTK_UNKNUSR	Yes/No	4	365	368	Is this an undefined user?
RALT_UTK_ERROR	Yes/No	4	370	373	Is this user token in error?
RALT_UTK_TRUSTED	Yes/No	4	375	378	Is this user a part of the trusted computing base (TCB)?
RALT_UTK_SESSTYPE	Char	8	380	387	The session type of this session.
RALT_UTK_SURROGAT	Yes/No	4	389	392	Is this a surrogate user?
RALT_UTK_REMOTE	Yes/No	4	394	397	Is this a remote job?
RALT_UTK_PRIV	Yes/No	4	399	402	Is this a privileged user ID?
RALT_UTK_SECL	Char	8	404	411	The security label of the user.
RALT_UTK_EXECNODE	Char	8	413	420	The execution node of the work.
RALT_UTK_SUSER_ID	Char	8	422	429	The submitting user ID.
RALT_UTK_SNODE	Char	8	431	438	The submitting node.
RALT_UTK_SGRP_ID	Char	8	440	447	The submitting group name.
RALT_UTK_SPOE	Char	8	449	456	The port of entry.
RALT_UTK_SPCLASS	Char	8	458	465	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
RALT_UTK_USER_ID	Char	8	467	474	User ID associated with the record.
RALT_UTK_GRP_ID	Char	8	476	483	Group name associated with the record.
RALT_UTK_DFT_GRP	Yes/No	4	485	488	Is a default group assigned?
RALT_UTK_DFT_SECL	Yes/No	4	490	493	Is a default security label assigned?
RALT_APPC_LINK	Char	16	495	510	Key to link together APPC records.
RALT_RES_NAME	Char	255	512	766	The resource name.
RALT_SPECIFIED	Char	1024	768	1791	The keywords specified.
RALT_FAILED	Char	1024	1793	2816	The keywords that failed.
RALT_UTK_NETW	Char	8	2818	2825	The port of entry network name.
RALT_X500_SUBJECT	Char	255	2827	3081	Subject's name associated with this event.
RALT_X500_ISSUER	Char	255	3083	3337	Issuer's name associated with this event.
RALT_SERV_POENAME	Char	64	3339	3402	SERVAUTH resource or profile name.
RALT_CTX_USER	Char	510	3404	3913	Authenticated user name.
RALT_CTX_REG	Char	255	3915	4169	Authenticated user registry name.

Table 48. Format of the RALTER record extension (event code 20) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RALT_CTX_HOST	Char	128	4171	4298	Authenticated user host name.
RALT_CTX_MECH	Char	16	4300	4315	Authenticated user authentication mechanism object identifier (OID).
RALT_IDID_USER	Char	985	4317	5301	Authenticated distributed user name.
RALT_IDID_REG	Char	1021	5303	6323	Authenticated distributed user registry name.

Table 49. Event qualifiers for RALTER command records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The RDEFINE record extension

Table 50 on page 214 describes the format of a record that is created by the RDEFINE command.

The event qualifiers that can be associated with an RDEFINE command are shown in Table 51 on page 215.

Table 50. Format of the RDEFINE record extension (event code 21)					
Field name	Type	Length	Position		Comments
			Start	End	
RDEF_CLASS	Char	8	282	289	Class name.
RDEF_OWN_ID	Char	8	291	298	Owner of the profile.
RDEF_USER_NAME	Char	20	300	319	User name.
RDEF_SECL	Char	8	321	328	The security label associated with the profile.
RDEF_UTK_ENCR	Yes/No	4	330	333	Is the UTKEN associated with this user encrypted?
RDEF_UTK_PRE19	Yes/No	4	335	338	Is this a pre-1.9 token?
RDEF_UTK_VERPROF	Yes/No	4	340	343	Is the VERIFYX propagation flag set?
RDEF_UTK_NJEUNUSR	Yes/No	4	345	348	Is this the NJE undefined user?
RDEF_UTK_LOGUSR	Yes/No	4	350	353	Is UAUDIT specified for this user?
RDEF_UTK_SPECIAL	Yes/No	4	355	358	Is this a SPECIAL user?
RDEF_UTK_DEFAULT	Yes/No	4	360	363	Is this a default token?
RDEF_UTK_UNKNUSR	Yes/No	4	365	368	Is this an undefined user?
RDEF_UTK_ERROR	Yes/No	4	370	373	Is this user token in error?
RDEF_UTK_TRUSTED	Yes/No	4	375	378	Is this user a part of the trusted computing base (TCB)?
RDEF_UTK_SESSTYPE	Char	8	380	387	The session type of this session.
RDEF_UTK_SURROGAT	Yes/No	4	389	392	Is this a surrogate user?
RDEF_UTK_REMOTE	Yes/No	4	394	397	Is this a remote job?
RDEF_UTK_PRIV	Yes/No	4	399	402	Is this a privileged user ID?
RDEF_UTK_SECL	Char	8	404	411	The security label of the user.

Table 50. Format of the RDEFINE record extension (event code 21) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RDEF_UTK_EXECNODE	Char	8	413	420	The execution node of the work.
RDEF_UTK_SUSER_ID	Char	8	422	429	The submitting user ID.
RDEF_UTK_SNODE	Char	8	431	438	The submitting node.
RDEF_UTK_SGRP_ID	Char	8	440	447	The submitting group name.
RDEF_UTK_SPOE	Char	8	449	456	The port of entry.
RDEF_UTK_SPCLASS	Char	8	458	465	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
RDEF_UTK_USER_ID	Char	8	467	474	User ID associated with the record.
RDEF_UTK_GRP_ID	Char	8	476	483	Group name associated with the record.
RDEF_UTK_DFT_GRP	Yes/No	4	485	488	Is a default group assigned?
RDEF_UTK_DFT_SECL	Yes/No	4	490	493	Is a default security label assigned?
RDEF_APPC_LINK	Char	16	495	510	Key to link together APPC records.
RDEF_RES_NAME	Char	255	512	766	The resource name.
RDEF_SPECIFIED	Char	1024	768	1791	The keywords specified.
RDEF_FAILED	Char	1024	1793	2816	The keywords that failed.
RDEF_UTK_NETW	Char	8	2818	2825	The port of entry network name.
RDEF_X500_SUBJECT	Char	255	2827	3081	Subject's name associated with this event.
RDEF_X500_ISSUER	Char	255	3083	3337	Issuer's name associated with this event.
RDEF_SERV_POENAME	Char	64	3339	3402	SERVAUTH resource or profile name.
RDEF_CTX_USER	Char	510	3404	3913	Authenticated user name.
RDEF_CTX_REG	Char	255	3915	4169	Authenticated user registry name.
RDEF_CTX_HOST	Char	128	4171	4298	Authenticated user host name.
RDEF_CTX_MECH	Char	16	4300	4315	Authenticated user authentication mechanism object identifier (OID).
RDEF_IDID_USER	Char	985	4317	5301	Authenticated distributed user name.
RDEF_IDID_REG	Char	1021	5303	6323	Authenticated distributed user registry name.

Table 51. Event qualifiers for RDEFINE command records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The RDELETE record extension

Table 52 on page 216 describes the format of a record that is created by the RDELETE command.

The event qualifiers that can be associated with an RDELETE command are shown in Table 53 on page 217.

Table 52. Format of the RDELETE record extension (event code 22)					
Field name	Type	Length	Position		Comments
			Start	End	
RDEL_CLASS	Char	8	282	289	Class name.
RDEL_OWN_ID	Char	8	291	298	Owner of the profile.
RDEL_USER_NAME	Char	20	300	319	User name.
RDEL_SECL	Char	8	321	328	The security label associated with the profile.
RDEL_UTK_ENCR	Yes/No	4	330	333	Is the UTOKEN associated with this user encrypted?
RDEL_UTK_PRE19	Yes/No	4	335	338	Is this a pre-1.9 token?
RDEL_UTK_VERPROF	Yes/No	4	340	343	Is the VERIFYX propagation flag set?
RDEL_UTK_NJEUNUSR	Yes/No	4	345	348	Is this the NJE undefined user?
RDEL_UTK_LOGUSR	Yes/No	4	350	353	Is UAUDIT specified for this user?
RDEL_UTK_SPECIAL	Yes/No	4	355	358	Is this a SPECIAL user?
RDEL_UTK_DEFAULT	Yes/No	4	360	363	Is this a default token?
RDEL_UTK_UNKNUSR	Yes/No	4	365	368	Is this an undefined user?
RDEL_UTK_ERROR	Yes/No	4	370	373	Is this user token in error?
RDEL_UTK_TRUSTED	Yes/No	4	375	378	Is this user a part of the trusted computing base (TCB)?
RDEL_UTK_SESTYPE	Char	8	380	387	The session type of this session.
RDEL_UTK_SURROGAT	Yes/No	4	389	392	Is this a surrogate user?
RDEL_UTK_REMOTE	Yes/No	4	394	397	Is this a remote job?
RDEL_UTK_PRIV	Yes/No	4	399	402	Is this a privileged user ID?
RDEL_UTK_SECL	Char	8	404	411	The security label of the user.
RDEL_UTK_EXECNODE	Char	8	413	420	The execution node of the work.
RDEL_UTK_SUSER_ID	Char	8	422	429	The submitting user ID.
RDEL_UTK_SNODE	Char	8	431	438	The submitting node.
RDEL_UTK_SGRP_ID	Char	8	440	447	The submitting group name.
RDEL_UTK_SPOE	Char	8	449	456	The port of entry.
RDEL_UTK_SPCLASS	Char	8	458	465	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
RDEL_UTK_USER_ID	Char	8	467	474	User ID associated with the record.
RDEL_UTK_GRP_ID	Char	8	476	483	Group name associated with the record.
RDEL_UTK_DFT_GRP	Yes/No	4	485	488	Is a default group assigned?
RDEL_UTK_DFT_SECL	Yes/No	4	490	493	Is a default security label assigned?
RDEL_APPC_LINK	Char	16	495	510	Key to link together APPC records.
RDEL_RES_NAME	Char	255	512	766	The resource name.
RDEL_SPECIFIED	Char	1024	768	1791	The keywords specified.
RDEL_UTK_NETW	Char	8	1793	1800	The port of entry network name.
RDEL_X500_SUBJECT	Char	255	1802	2056	Subject's name associated with this event.
RDEL_X500_ISSUER	Char	255	2058	2312	Issuer's name associated with this event.
RDEL_SERV_POENAME	Char	64	2314	2377	SERVAUTH resource or profile name.
RDEL_CTX_USER	Char	510	2379	2888	Authenticated user name.

Table 52. Format of the RDELETE record extension (event code 22) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RDEL_CTX_REG	Char	255	2890	3144	Authenticated user registry name.
RDEL_CTX_HOST	Char	128	3146	3273	Authenticated user host name.
RDEL_CTX_MECH	Char	16	3275	3290	Authenticated user authentication mechanism object identifier (OID).
RDEL_IDID_USER	Char	985	3292	4276	Authenticated distributed user name.
RDEL_IDID_REG	Char	1021	4278	5298	Authenticated distributed user registry name.

Table 53. Event qualifiers for RDELETE command records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The REMOVE record extension

Table 54 on page 217 describes the format of a record that is created by the REMOVE command.

The event qualifiers that can be associated with a REMOVE command are shown in Table 55 on page 218.

Table 54. Format of the REMOVE record extension (event code 23)					
Field name	Type	Length	Position		Comments
			Start	End	
REM_OWN_ID	Char	8	282	289	Owner of the profile.
REM_USER_NAME	Char	20	291	310	User name.
REM_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
REM_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
REM_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
REM_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
REM_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
REM_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
REM_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
REM_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
REM_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
REM_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
REM_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
REM_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
REM_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
REM_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
REM_UTK_SECL	Char	8	386	393	The security label of the user.
REM_UTK_EXECNODE	Char	8	395	402	The execution node of the work.

Table 54. Format of the REMOVE record extension (event code 23) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
REM_UTK_USUSER_ID	Char	8	404	411	The submitting user ID.
REM_UTK_SNODE	Char	8	413	420	The submitting node.
REM_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
REM_UTK_SPOE	Char	8	431	438	The port of entry.
REM_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
REM_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
REM_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
REM_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
REM_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
REM_APPC_LINK	Char	16	477	492	Key to link together APPC records.
REM_USER_ID	Char	8	494	501	The user ID.
REM_SPECIFIED	Char	1024	503	1526	The keywords specified.
REM_FAILED	Char	1024	1528	2551	The keywords that failed.
REM_UTK_NETW	Char	8	2553	2560	The port of entry network name.
REM_X500_SUBJECT	Char	255	2562	2816	Subject's name associated with this event.
REM_X500_ISSUER	Char	255	2818	3072	Issuer's name associated with this event.
REM_SERV_POENAME	Char	64	3074	3137	SERVAUTH resource or profile name.
REM_CTX_USER	Char	510	3139	3648	Authenticated user name.
REM_CTX_REG	Char	255	3650	3904	Authenticated user registry name.
REM_CTX_HOST	Char	128	3906	4033	Authenticated user host name.
REM_CTX_MECH	Char	16	4035	4050	Authenticated user authentication mechanism object identifier (OID).
REM_IDID_USER	Char	985	4052	5036	Authenticated distributed user name.
REM_IDID_REG	Char	1021	5038	6058	Authenticated distributed user registry name.

Table 55. Event qualifiers for REMOVE command records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The SETROPTS record extension

Table 56 on page 219 describes record format that is created by the SETROPTS command.

Table 57 on page 220 shows the event qualifiers that can be associated with a SETROPTS command.

Table 56. Format of the SETROPTS record extension (event code 24)					
Field name	Type	Length	Position		Comments
			Start	End	
SETR_USER_NAME	Char	20	282	301	User name.
SETR_UTK_ENCR	Yes/No	4	303	306	Is the UTOKEN associated with this user encrypted?
SETR_UTK_PRE19	Yes/No	4	308	311	Is this a pre-1.9 token?
SETR_UTK_VERPROF	Yes/No	4	313	316	Is the VERIFYX propagation flag set?
SETR_UTK_NJEUNUSR	Yes/No	4	318	321	Is this the NJE undefined user?
SETR_UTK_LOGUSR	Yes/No	4	323	326	Is UAUDIT specified for this user?
SETR_UTK_SPECIAL	Yes/No	4	328	331	Is this a SPECIAL user?
SETR_UTK_DEFAULT	Yes/No	4	333	336	Is this a default token?
SETR_UTK_UNKNUSR	Yes/No	4	338	341	Is this an undefined user?
SETR_UTK_ERROR	Yes/No	4	343	346	Is this user token in error?
SETR_UTK_TRUSTED	Yes/No	4	348	351	Is this user a part of the trusted computing base (TCB)?
SETR_UTK_SESTYPE	Char	8	353	360	The session type of this session.
SETR_UTK_SURROGAT	Yes/No	4	362	365	Is this a surrogate user?
SETR_UTK_REMOTE	Yes/No	4	367	370	Is this a remote job?
SETR_UTK_PRIV	Yes/No	4	372	375	Is this a privileged user ID?
SETR_UTK_SECL	Char	8	377	384	The security label of the user.
SETR_UTK_EXECNODE	Char	8	386	393	The execution node of the work.
SETR_UTK_SUSER_ID	Char	8	395	402	The submitting user ID.
SETR_UTK_SNODE	Char	8	404	411	The submitting node.
SETR_UTK_SGRP_ID	Char	8	413	420	The submitting group name.
SETR_UTK_SPOE	Char	8	422	429	The port of entry.
SETR_UTK_SPCCLASS	Char	8	431	438	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
SETR_UTK_USER_ID	Char	8	440	447	User ID associated with the record.
SETR_UTK_GRP_ID	Char	8	449	456	Group name associated with the record.
SETR_UTK_DFT_GRP	Yes/No	4	458	461	Is a default group assigned?
SETR_UTK_DFT_SECL	Yes/No	4	463	466	Is a default security label assigned?
SETR_APPC_LINK	Char	16	468	483	Key to link together APPC records.
SETR_SPECIFIED	Char	1024	485	1508	The keywords specified.
SETR_FAILED	Char	1024	1510	2533	The keywords that failed.
SETR_UTK_NETW	Char	8	2535	2542	The port of entry network name.
SETR_X500_SUBJECT	Char	255	2544	2798	Subject's name associated with this event.
SETR_X500_ISSUER	Char	255	2800	3054	Issuer's name associated with this event.
SETR_SERV_POENAME	Char	64	3056	3119	SERVAUTH resource or profile name.
SETR_CTX_USER	Char	510	3121	3630	Authenticated user name.
SETR_CTX_REG	Char	255	3632	3886	Authenticated user registry name.
SETR_CTX_HOST	Char	128	3888	4015	Authenticated user host name.
SETR_CTX_MECH	Char	16	4017	4032	Authenticated user authentication mechanism object identifier (OID).

Table 56. Format of the SETROPTS record extension (event code 24) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
SETR_IDID_USER	Char	985	4034	5018	Authenticated distributed user name.
SETR_IDID_REG	Char	1021	5020	6040	Authenticated distributed user registry name.

Table 57. Event qualifiers for SETROPTS command records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The RVARY record extension

Table 58 on page 220 describes the format of a record that is created by the RVARY command.

The event qualifiers that can be associated with an RVARY command are shown in Table 59 on page 221.

Table 58. Format of the RVARY record extension (event code 25)					
Field name	Type	Length	Position		Comments
			Start	End	
RVAR_USER_NAME	Char	20	282	301	User name.
RVAR_UTK_ENCR	Yes/No	4	303	306	Is the UTOKEN associated with this user encrypted?
RVAR_UTK_PRE19	Yes/No	4	308	311	Is this a pre-1.9 token?
RVAR_UTK_VERPROF	Yes/No	4	313	316	Is the VERIFYX propagation flag set?
RVAR_UTK_NJEUNUSR	Yes/No	4	318	321	Is this the NJE undefined user?
RVAR_UTK_LOGUSR	Yes/No	4	323	326	Is UAUDIT specified for this user?
RVAR_UTK_SPECIAL	Yes/No	4	328	331	Is this a SPECIAL user?
RVAR_UTK_DEFAULT	Yes/No	4	333	336	Is this a default token?
RVAR_UTK_UNKNUSR	Yes/No	4	338	341	Is this an undefined user?
RVAR_UTK_ERROR	Yes/No	4	343	346	Is this user token in error?
RVAR_UTK_TRUSTED	Yes/No	4	348	351	Is this user a part of the trusted computing base (TCB)?
RVAR_UTK_SESSTYPE	Char	8	353	360	The session type of this session.
RVAR_UTK_SURROGAT	Yes/No	4	362	365	Is this a surrogate user?
RVAR_UTK_REMOTE	Yes/No	4	367	370	Is this a remote job?
RVAR_UTK_PRIV	Yes/No	4	372	375	Is this a privileged user ID?
RVAR_UTK_SECL	Char	8	377	384	The security label of the user.
RVAR_UTK_EXECNODE	Char	8	386	393	The execution node of the work.
RVAR_UTK_SUSER_ID	Char	8	395	402	The submitting user ID.
RVAR_UTK_SNODE	Char	8	404	411	The submitting node.
RVAR_UTK_SGRP_ID	Char	8	413	420	The submitting group name.
RVAR_UTK_SPOE	Char	8	422	429	The port of entry.

Table 58. Format of the RVAR record extension (event code 25) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RVAR_UTK_SPCCLASS	Char	8	431	438	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
RVAR_UTK_USER_ID	Char	8	440	447	User ID associated with the record.
RVAR_UTK_GRP_ID	Char	8	449	456	Group name associated with the record.
RVAR_UTK_DFT_GRP	Yes/No	4	458	461	Is a default group assigned?
RVAR_UTK_DFT_SECL	Yes/No	4	463	466	Is a default security label l assigned?
RVAR_APPC_LINK	Char	16	468	483	Key to link together APPC records.
RVAR_SPECIFIED	Char	1024	485	1508	The keywords specified.
RVAR_FAILED	Char	1024	1510	2533	The keywords that failed.
RVAR_UTK_NETW	Char	8	2535	2542	The port of entry network name.
RVAR_X500_SUBJECT	Char	255	2544	2798	Subject's name associated with this event.
RVAR_X500_ISSUER	Char	255	2800	3054	Issuer's name associated with this event.
RVAR_SERV_POENAME	Char	64	3056	3119	SERVAUTH resource or profile name.
RVAR_CTX_USER	Char	510	3121	3630	Authenticated user name.
RVAR_CTX_REG	Char	255	3632	3886	Authenticated user registry name.
RVAR_CTX_HOST	Char	128	3888	4015	Authenticated user host name.
RVAR_CTX_MECH	Char	16	4017	4032	Authenticated user authentication mechanism object identifier (OID).
RVAR_IDID_USER	Char	985	4034	5018	Authenticated distributed user name.
RVAR_IDID_REG	Char	1021	5020	6040	Authenticated distributed user registry name.

Table 59. Event qualifiers for RVAR command records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The APPCLU record extension

Table 60 on page 221 describes the format of a record that is created by the auditing of an APPCLU resource.

The event qualifiers that can be associated with an APPCLU (APPC session establishment) event are shown in Table 61 on page 223.

Table 60. Format of the APPCLU record extension (event code 26)					
Field name	Type	Length	Position		Comments
			Start	End	
APPC_RES_NAME	Char	255	282	536	Resource name.
APPC_CLASS	Char	8	538	545	Class name.
APPC_TYPE	Char	8	547	554	Type of resource data. Valid values are "RESOURCE" if ACC_NAME is a generic resource name, and "PROFILE" if ACC_NAME is a generic profile.

Table 60. Format of the APPCLU record extension (event code 26) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
APPC_NAME	Char	246	556	801	Resource or profile name.
APPC_OWN_ID	Char	8	803	810	Name of the profile owner.
APPC_USER_NAME	Char	20	812	831	User name.
APPC_UTK_ENCR	Yes/No	4	833	836	Is the UTOKEN associated with this user encrypted?
APPC_UTK_PRE19	Yes/No	4	838	841	Is this a pre-1.9 token?
APPC_UTK_VERPROF	Yes/No	4	843	846	Is the VERIFYX propagation flag set?
APPC_UTK_NJEUNUSR	Yes/No	4	848	851	Is this the NJE undefined user?
APPC_UTK_LOGUSR	Yes/No	4	853	856	Is UAUDIT specified for this user?
APPC_UTK_SPECIAL	Yes/No	4	858	861	Is this a SPECIAL user?
APPC_UTK_DEFAULT	Yes/No	4	863	866	Is this a default token?
APPC_UTK_UNKNUSR	Yes/No	4	868	871	Is this an undefined user?
APPC_UTK_ERROR	Yes/No	4	873	876	Is this user token in error?
APPC_UTK_TRUSTED	Yes/No	4	878	881	Is this user a part of the trusted computing base (TCB)?
APPC_UTK_SESSTYPE	Char	8	883	890	The session type of this session.
APPC_UTK_SURROGAT	Yes/No	4	892	895	Is this a surrogate user?
APPC_UTK_REMOTE	Yes/No	4	897	900	Is this a remote job?
APPC_UTK_PRIV	Yes/No	4	902	905	Is this a privileged user ID?
APPC_UTK_SECL	Char	8	907	914	The security label of the user.
APPC_UTK_EXECNODE	Char	8	916	923	The execution node of the work.
APPC_UTK_SUSER_ID	Char	8	925	932	The submitting user ID.
APPC_UTK_SNODE	Char	8	934	941	The submitting node.
APPC_UTK_SGRP_ID	Char	8	943	950	The submitting group name.
APPC_UTK_SPOE	Char	8	952	959	The port of entry.
APPC_UTK_SPCCLASS	Char	8	961	968	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
APPC_UTK_USER_ID	Char	8	970	977	User ID associated with the record.
APPC_UTK_GRP_ID	Char	8	979	986	Group name associated with the record.
APPC_UTK_DFT_GRP	Yes/No	4	988	991	Is a default group assigned?
APPC_UTK_DFT_SECL	Yes/No	4	993	996	Is a default security label assigned?
APPC_APPC_LINK	Char	16	998	1013	Key to link together APPC records.
APPC_UTK_NETW	Char	8	1015	1022	The port of entry network name.
APPC_X500_SUBJECT	Char	255	1024	1278	Subject's name associated with this event.
APPC_X500_ISSUER	Char	255	1280	1534	Issuer's name associated with this event.
APPC_SERV_POENAME	Char	64	1536	1599	SERVAUTH resource or profile name.
APPC_CTX_USER	Char	510	1601	2110	Authenticated user name.
APPC_CTX_REG	Char	255	2112	2366	Authenticated user registry name.
APPC_CTX_HOST	Char	128	2368	2495	Authenticated user host name.
APPC_CTX_MECH	Char	16	2497	2512	Authenticated user authentication mechanism object identifier (OID).

Table 60. Format of the APPCLU record extension (event code 26) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
APPC_IDID_USER	Char	985	2514	3498	Authenticated distributed user name.
APPC_IDID_REG	Char	1021	3500	4520	Authenticated distributed user registry name.

Table 61. Event qualifiers for APPC session establishment records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Partner verification OK.
NOVERIFY	01	Session established without verification.
LKEYEXPR	02	Local key expires in less than 5 days.
REVOKED	03	Partner LU access has been revoked.
NOMATCH	04	Partner LU key does not match this LU key.
TRMSECUR	05	Session terminated for security reasons.
NOSESKEY	06	Required session key not defined.
LUATTACK	07	Possible security attack by partner LU.
NOPRTKEY	08	Session key not defined for the partner LU.
NOKEY	09	Session key not defined for this LU.
SNAERROR	10	SNA security-related session
PROFCHNG	11	Profile changed during verification.
SKEYEXPR	12	Expired session key error.

The general event record extension

Table 62 on page 223 describes the format of a record that is created by a general event.

The event qualifiers that can be associated with a general event are determined by the installation. These event codes are unloaded as integer values.

Table 62. Format of the general event record extension (event code 27)					
Field name	Type	Length	Position		Comments
			Start	End	
GEN_CLASS	Char	8	282	289	Class name.
GEN_LOGSTR	Char	255	291	545	LOGSTR= data from the RACROUTE
GEN_USER_NAME	Char	20	547	566	User name.
GEN_UTK_ENCR	Yes/No	4	568	571	Is the UTOKEN associated with this user encrypted?
GEN_UTK_PRE19	Yes/No	4	573	576	Is this a pre-1.9 token?
GEN_UTK_VERPROF	Yes/No	4	578	581	Is the VERIFYX propagation flag set?
GEN_UTK_NJEUNUSR	Yes/No	4	583	586	Is this the NJE undefined user?
GEN_UTK_LOGUSR	Yes/No	4	588	591	Is UAUDIT specified for this user?
GEN_UTK_SPECIAL	Yes/No	4	593	596	Is this a SPECIAL user?
GEN_UTK_DEFAULT	Yes/No	4	598	601	Is this a default token?
GEN_UTK_UNKNUSR	Yes/No	4	603	606	Is this an undefined user?
GEN_UTK_ERROR	Yes/No	4	608	611	Is this user token in error?

Table 62. Format of the general event record extension (event code 27) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
GEN_UTK_TRUSTED	Yes/No	4	613	616	Is this user a part of the trusted computing base (TCB)?
GEN_UTK_SESSTYPE	Char	8	618	625	The session type of this session.
GEN_UTK_SURROGAT	Yes/No	4	627	630	Is this a surrogate user?
GEN_UTK_REMOTE	Yes/No	4	632	635	Is this a remote job?
GEN_UTK_PRIV	Yes/No	4	637	640	Is this a privileged user ID?
GEN_UTK_SECL	Char	8	642	649	The security label of the user.
GEN_UTK_EXECNODE	Char	8	651	658	The execution node of the work.
GEN_UTK_SUSER_ID	Char	8	660	667	The submitting user ID.
GEN_UTK_SNODE	Char	8	669	676	The submitting node.
GEN_UTK_SGRP_ID	Char	8	678	685	The submitting group name.
GEN_UTK_SPOE	Char	8	687	694	The port of entry.
GEN_UTK_SPCLASS	Char	8	696	703	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
GEN_UTK_USER_ID	Char	8	705	712	User ID associated with the record.
GEN_UTK_GRP_ID	Char	8	714	721	Group name associated with the record.
GEN_UTK_DFT_GRP	Yes/No	4	723	726	Is a default group assigned?
GEN_UTK_DFT_SECL	Yes/No	4	728	731	Is a default security label assigned?
GEN_APPC_LINK	Char	16	733	748	Key to link together GENERAL records.
GEN_UTK_NETW	Char	8	750	757	The port of entry network name.
GEN_X500_SUBJECT	Char	255	759	1013	Subject's name associated with this event.
GEN_X500_ISSUER	Char	255	1015	1269	Issuer's name associated with this event.
GEN_SERV_POENAME	Char	64	1271	1334	SERVAUTH resource or profile name.
GEN_CTX_USER	Char	510	1336	1845	Authenticated user name.
GEN_CTX_REG	Char	255	1847	2101	Authenticated user registry name.
GEN_CTX_HOST	Char	128	2103	2230	Authenticated user host name.
GEN_CTX_MECH	Char	16	2232	2247	Authenticated user authentication mechanism object identifier (OID).
GEN_IDID_USER	Char	985	2249	3233	Authenticated distributed user name.
GEN_IDID_REG	Char	1021	3235	4255	Authenticated distributed user registry name.

The directory search record extension

Table 63 on page 224 describes the format of a record that is created by a directory search event.

The event qualifiers that can be associated with a directory search event are shown in Table 64 on page 227.

Table 63. Format of the directory search record extension (event code 28)					
Field name	Type	Length	Position		Comments
			Start	End	
DSCH_CLASS	Char	8	282	289	Class name.
DSCH_USER_NAME	Char	20	291	310	The name associated with the user ID.

Table 63. Format of the directory search record extension (event code 28) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
DSCH_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
DSCH_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
DSCH_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
DSCH_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
DSCH_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
DSCH_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
DSCH_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
DSCH_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
DSCH_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
DSCH_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
DSCH_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
DSCH_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
DSCH_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
DSCH_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
DSCH_UTK_SECL	Char	8	386	393	The security label of the user.
DSCH_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
DSCH_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
DSCH_UTK_SNODE	Char	8	413	420	The submitting node.
DSCH_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
DSCH_UTK_SPOE	Char	8	431	438	The port of entry.
DSCH_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DSCH_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
DSCH_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
DSCH_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
DSCH_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
DSCH_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
DSCH_AUDIT_CODE	Char	11	494	504	Audit function code.
DSCH_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
DSCH_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
DSCH_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
DSCH_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
DSCH_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
DSCH_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
DSCH_PATH_NAME	Char	1023	572	1594	The requested path name.
DSCH_FILE_ID	Char	32	1596	1627	File ID.
DSCH_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.

Table 63. Format of the directory search record extension (event code 28) (continued)

Field name	Type	Length	Position		Comments
			Start	End	
DSCH_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
DSCH_REQUEST_READ	Yes/No	4	1651	1654	Did the requested access include read?
DSCH_REQUEST_WRITE	Yes/No	4	1656	1659	Did the requested access include write?
DSCH_REQUEST_EXEC	Yes/No	4	1661	1664	Did the requested access include EXECUTE?
DSCH_REQUEST_DSRCH	Yes/No	4	1666	1669	Did the requested access include directory search?
DSCH_ACCESS_TYPE	Char	8	1671	1678	What bits were used in granting the access? Valid values are "ACLERROR", "ACLGROUP", "ACLUSER", "FSACCESS", "GROUP", "NO", "OTHER", "OWNER", and "RSTD".
DSCH_ALLOWED_READ	Yes/No	4	1680	1683	Was read access allowed?
DSCH_ALLOWED_WRITE	Yes/No	4	1685	1688	Was write access allowed?
DSCH_ALLOWED_EXEC	Yes/No	4	1690	1693	Was execute or search access allowed?
DSCH_REQUEST_PATH2	Char	1023	1695	2717	Second requested path name.
DSCH_SERVICE_CODE	Char	11	2719	2729	The service that was being processed. This is set only when the DSCH_AUDIT_CODE is "LOOKUP".
DSCH_HFS_DS_NAME	Char	44	2731	2774	Data set name for the mounted file system.
DSCH_SYMLINK	Char	1023	2776	3798	The content of SYMLINK.
DSCH_FILE_NAME	Char	256	3800	4055	The file name that is being checked.
DSCH_PATH_TYPE	Char	4	4057	4060	Type of the requested path name. Valid values are "OLD" and "NEW".
DSCH_FILEPOOL	Char	8	4062	4069	SFS filepool containing the BFS file.
DSCH_FILESSPACE	Char	8	4071	4078	SFS filespace containing the BFS file.
DSCH_INODE	Integer	10	4080	4089	Inode (file serial number).
DSCH_SCID	Integer	10	4091	4100	File SCID.
DSCH_DCE_LINK	Char	16	4102	4117	Link to connect DCE records that originate from a single DCE request.
DSCH_AUTH_TYPE	Char	13	4119	4131	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
DSCH_DFLT_PROCESS	Yes/No	4	4133	4136	Default z/OS UNIX security environment in effect.
DSCH_UTK_NETW	CHAR	8	4138	4145	The port of entry network name.
DSCH_X500_SUBJECT	Char	255	4147	4401	Subject's name associated with this event.
DSCH_X500_ISSUER	Char	255	4403	4657	Issuer's name associated with this event.
DSCH_SECL	Char	8	4659	4666	Security label of the resource.
DSCH_SERV_POENAME	Char	64	4668	4731	SERVAUTH resource or profile name.
DSCH_CTX_USER	Char	510	4733	5242	Authenticated user name.
DSCH_CTX_REG	Char	255	5244	5498	Authenticated user registry name.
DSCH_CTX_HOST	Char	128	5500	5627	Authenticated user host name.
DSCH_CTX_MECH	Char	16	5629	5644	Authenticated user authentication mechanism object identifier (OID).
DSCH_IDID_USER	Char	985	5646	6630	Authenticated distributed user name.
DSCH_IDID_REG	Char	1021	6632	7652	Authenticated distributed user registry name.

Table 64. Event qualifiers for directory search records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	Access allowed.
NOTAUTH	01	Not authorized to search the directory.
INSSECL	02	Insufficient security label.

The check directory access record extension

Table 65 on page 227 describes the format of a record that is created by checking access to a directory.

The event qualifiers that can be associated with a directory search event are shown in Table 66 on page 229.

Table 65. Format of the check directory access record extension (event code 29)

Field name	Type	Length	Position		Comments
			Start	End	
DACC_CLASS	Char	8	282	289	Class name.
DACC_USER_NAME	Char	20	291	310	The name associated with the user ID.
DACC_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
DACC_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
DACC_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
DACC_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
DACC_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
DACC_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
DACC_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
DACC_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
DACC_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
DACC_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
DACC_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
DACC_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
DACC_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
DACC_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
DACC_UTK_SECL	Char	8	386	393	The security label of the user.
DACC_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
DACC_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
DACC_UTK_SNODE	Char	8	413	420	The submitting node.
DACC_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
DACC_UTK_SPOE	Char	8	431	438	The port of entry.
DACC_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DACC_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
DACC_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
DACC_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?

Table 65. Format of the check directory access record extension (event code 29) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
DACC_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
DACC_APPC_LINK	Char	16	477	492	Key to link together APPC records.
DACC_AUDIT_CODE	Char	11	494	504	Audit function code.
DACC_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
DACC_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
DACC_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
DACC_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
DACC_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
DACC_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
DACC_PATH_NAME	Char	1023	572	1594	The requested path name.
DACC_FILE_ID	Char	32	1596	1627	File ID.
DACC_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
DACC_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
DACC_REQUEST_READ	Yes/No	4	1651	1654	Did the requested access include read?
DACC_REQUEST_WRITE	Yes/No	4	1656	1659	Did the requested access include write?
DACC_REQUEST_EXEC	Yes/No	4	1661	1664	Did the requested access include execute?
DACC_REQUEST_DSRCH	Yes/No	4	1666	1669	Did the requested access include directory search?
DACC_ACCESS_TYPE	Char	8	1671	1678	What bits were used in granting the access? Valid values are "OWNER", "GROUP", "OTHER", "ACLUER", "ACLGROUP", "ACLERROR", "RSTD", and "NO".
DACC_ALLOWED_READ	Yes/No	4	1680	1683	Was read access allowed?
DACC_ALLOWED_WRITE	Yes/No	4	1685	1688	Was write access allowed?
DACC_ALLOWED_EXEC	Yes/No	4	1690	1693	Was execute access allowed?
DACC_REQUEST_PATH2	Char	1023	1695	2717	Second requested path name.
DACC_SYMLINK	Char	1023	2719	3741	The content of SYMLINK.
DACC_FILE_NAME	Char	256	3743	3998	The file name that is being checked.
DACC_PATH_TYPE	Char	4	4000	4003	Type of the requested path name. Valid values are "OLD" and "NEW".
DACC_FILEPOOL	Char	8	4005	4012	SFS filepool containing the BFS file.
DACC_FILESPEACE	Char	8	4014	4021	SFS filespace containing the BFS file.
DACC_INODE	Integer	10	4023	4032	Inode (file serial number).
DACC_SCID	Integer	10	4034	4043	File SCID.
DACC_DCE_LINK	Char	16	4045	4060	Link to connect DCE records that originate from a single DCE request.
DACC_AUTH_TYPE	Char	13	4062	4074	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
DACC_DFLT_PROCESS	Yes/No	4	4076	4079	Default z/OS UNIX security environment in effect.
DACC_UTK_NETW	Char	8	4081	4088	The port of entry network name.
DACC_X500_SUBJECT	Char	255	4090	4344	Subject's name associated with this event.

Table 65. Format of the check directory access record extension (event code 29) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
DACC_X500_ISSUER	Char	255	4346	4600	Issuer's name associated with this event.
DACC_SECL	Char	8	4602	4609	Security label of the resource.
DACC_SERV_POENAME	Char	64	4611	4674	SERVAUTH resource or profile name.
DACC_CTX_USER	Char	510	4676	5185	Authenticated user name.
DACC_CTX_REG	Char	255	5187	5441	Authenticated user registry name.
DACC_CTX_HOST	Char	128	5443	5570	Authenticated user host name.
DACC_CTX_MECH	Char	16	5572	5587	Authenticated user authentication mechanism object identifier (OID).
DACC_IDID_USER	Char	985	5589	6573	Authenticated distributed user name.
DACC_IDID_REG	Char	1021	6575	7595	Authenticated distributed user registry name.

Table 66. Event qualifiers for check directory records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Access allowed.
NOTAUTH	01	Not authorized to the directory.
INSSECL	02	Insufficient security label.

The check file access record extension

Table 67 on page 229 describes the format of a record that is created by checking access to a file.

The event qualifiers that can be associated with a check file access event are shown in Table 68 on page 231.

Table 67. Format of the check file access record extension (event code 30)					
Field name	Type	Length	Position		Comments
			Start	End	
FACC_CLASS	Char	8	282	289	Class name.
FACC_USER_NAME	Char	20	291	310	The name associated with the user ID.
FACC_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
FACC_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
FACC_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
FACC_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
FACC_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
FACC_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
FACC_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
FACC_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
FACC_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
FACC_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
FACC_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
FACC_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?

Table 67. Format of the check file access record extension (event code 30) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
FACC_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
FACC_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
FACC_UTK_SECL	Char	8	386	393	The security label of the user.
FACC_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
FACC_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
FACC_UTK_SNODE	Char	8	413	420	The submitting node.
FACC_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
FACC_UTK_SPOE	Char	8	431	438	The port of entry.
FACC_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
FACC_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
FACC_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
FACC_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
FACC_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
FACC_APPC_LINK	Char	16	477	492	Key to link together APPC records.
FACC_AUDIT_CODE	Char	11	494	504	Audit function code.
FACC_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
FACC_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
FACC_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
FACC_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
FACC_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
FACC_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
FACC_PATH_NAME	Char	1023	572	1594	The requested path name.
FACC_FILE_ID	Char	32	1596	1627	File ID.
FACC_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
FACC_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
FACC_REQUEST_READ	Yes/No	4	1651	1654	Did the requested access include read?
FACC_REQUEST_WRITE	Yes/No	4	1656	1659	Did the requested access include write?
FACC_REQUEST_EXEC	Yes/No	4	1661	1664	Did the requested access include EXECUTE?
FACC_REQUEST_DSRCH	Yes/No	4	1666	1669	Did the requested access include directory search?
FACC_ACCESS_TYPE	Char	8	1671	1678	What bits were used in granting the access? Valid values are "OWNER", "GROUP", "OTHER", "ACLSUSER", "ACLGROUP", "ACLERROR", "RSTD", and "NO".
FACC_ALLOWED_READ	Yes/No	4	1680	1683	Was read access allowed?
FACC_ALLOWED_WRITE	Yes/No	4	1685	1688	Was write access allowed?
FACC_ALLOWED_EXEC	Yes/No	4	1690	1693	Was execute access allowed?
FACC_REQUEST_PATH2	Char	1023	1695	2717	Second requested path name.
FACC_FILE_NAME	Char	256	2719	2974	The file name that is being checked.

Table 67. Format of the check file access record extension (event code 30) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
FACC_PATH_TYPE	Char	4	2976	2979	Type of the requested path name. Valid values are "OLD" and "NEW".
FACC_FILEPOOL	Char	8	2981	2988	SFS filepool containing the BFS file.
FACC_FILESPACE	Char	8	2990	2997	SFS filespace containing the BFS file.
FACC_INODE	Integer	10	2999	3008	Inode (file serial number).
FACC_SCID	Integer	10	3010	3019	File SCID.
FACC_DCE_LINK	Char	16	3021	3036	Link to connect DCE records that originate from a single DCE request.
FACC_AUTH_TYPE	Char	13	3038	3050	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
FACC_DFLT_PROCESS	Yes/No	4	3052	3055	Default z/OS UNIX security environment in effect.
FACC_UTK_NETW	Char	8	3057	3064	The port of entry network name.
FACC_X500_SUBJECT	Char	255	3066	3320	Subject's name associated with this event.
FACC_X500_ISSUER	Char	255	3322	3576	Issuer's name associated with this event.
FACC_SECL	Char	8	3578	3585	Security label of the resource.
FACC_SERV_POENAME	Char	64	3587	3650	SERVAUTH resource or profile name.
FACC_CTX_USER	Char	510	3652	4161	Authenticated user name.
FACC_CTX_REG	Char	255	4163	4417	Authenticated user registry name.
FACC_CTX_HOST	Char	128	4419	4546	Authenticated user host name.
FACC_CTX_MECH	Char	16	4548	4563	Authenticated user authentication mechanism object identifier (OID).
FACC_IDID_USER	Char	985	4565	5549	Authenticated distributed user name.
FACC_IDID_REG	Char	1021	5551	6571	Authenticated distributed user registry name.

Table 68. Event qualifiers for check file access records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Access allowed.
NOTAUTH	01	Not authorized to the file.
INSSECL	02	Insufficient security label.

The change audit record extension

Table 69 on page 231 describes the format of a record that is created by checking access to a file.

The event qualifiers that can be associated with a directory search event are shown in Table 70 on page 234.

Table 69. Format of the change audit record extension (event code 31)					
Field name	Type	Length	Position		Comments
			Start	End	
CAUD_CLASS	Char	8	282	289	Class name.
CAUD_USER_NAME	Char	20	291	310	The name associated with the user ID.

Table 69. Format of the change audit record extension (event code 31) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
CAUD_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
CAUD_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
CAUD_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
CAUD_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
CAUD_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
CAUD_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
CAUD_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
CAUD_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
CAUD_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
CAUD_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
CAUD_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
CAUD_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
CAUD_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
CAUD_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
CAUD_UTK_SECL	Char	8	386	393	The security label of the user.
CAUD_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
CAUD_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
CAUD_UTK_SNODE	Char	8	413	420	The submitting node.
CAUD_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
CAUD_UTK_SPOE	Char	8	431	438	The port of entry.
CAUD_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
CAUD_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
CAUD_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
CAUD_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
CAUD_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
CAUD_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
CAUD_AUDIT_CODE	Char	11	494	504	Audit function code.
CAUD_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
CAUD_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
CAUD_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
CAUD_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
CAUD_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
CAUD_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
CAUD_PATH_NAME	Char	1023	572	1594	The requested path name.
CAUD_FILE_ID	Char	32	1596	1627	File ID.
CAUD_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.

Table 69. Format of the change audit record extension (event code 31) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
CAUD_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
CAUD_REQUEST_READ	Char	8	1651	1658	What audit options are requested for a READ operation? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_REQUEST_WRITE	Char	8	1660	1667	What audit options are requested for a WRITE operation? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_REQUEST_EXEC	Char	8	1669	1676	What audit options are requested for an EXECUTE operation? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_UOLD_READ	Char	8	1678	1685	What were the previous user audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_UOLD_WRITE	Char	8	1687	1694	What were the previous user audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_UOLD_EXEC	Char	8	1696	1703	What were the previous user audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_AOLD_READ	Char	8	1705	1712	What were the previous auditor audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_AOLD_WRITE	Char	8	1714	1721	What were the previous auditor audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_AOLD_EXEC	Char	8	1723	1730	What were the previous auditor audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_UNEW_READ	Char	8	1732	1739	What are the new user audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_UNEW_WRITE	Char	8	1741	1748	What are the new user audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_UNEW_EXEC	Char	8	1750	1757	What are the new user audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_ANEW_READ	Char	8	1759	1766	What are the new auditor audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_ANEW_WRITE	Char	8	1768	1775	What are the new auditor audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_ANEW_EXEC	Char	8	1777	1784	What are the new auditor audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_FILEPOOL	Char	8	1786	1793	SFS filepool containing the BFS file.
CAUD_FILESPACE	Char	8	1795	1802	SFS filespace containing the BFS file.
CAUD_INODE	Integer	10	1804	1813	Inode (file serial number).
CAUD_SCID	Integer	10	1815	1824	File SCID.

Table 69. Format of the change audit record extension (event code 31) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
CAUD_DCE_LINK	Char	16	1826	1841	Link to connect DCE records that originate from a single DCE request.
CAUD_AUTH_TYPE	Char	13	1843	1855	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
CAUD_DFLT_PROCESS	Yes/No	4	1857	1860	Default z/OS UNIX security environment in effect.
CAUD_UTK_NETW	Char	8	1862	1869	The port of entry network name.
CAUD_X500_SUBJECT	Char	255	1871	2125	Subject's name associated with this event.
CAUD_X500_ISSUER	Char	255	2127	2381	Issuer's name associated with this event.
CAUD_SECL	Char	8	2383	2390	Security label of the resource.
CAUD_SERV_POENAME	Char	64	2392	2455	SERVAUTH resource or profile name.
CAUD_CTX_USER	Char	510	2457	2966	Authenticated user name.
CAUD_CTX_REG	Char	255	2968	3222	Authenticated user registry name.
CAUD_CTX_HOST	Char	128	3224	3351	Authenticated user host name.
CAUD_CTX_MECH	Char	16	3353	3368	Authenticated user authentication mechanism object identifier (OID).
CAUD_IDID_USER	Char	985	3370	4354	Authenticated distributed user name.
CAUD_IDID_REG	Char	1021	4556	5376	Authenticated distributed user registry name.

Table 70. Event qualifiers for change audit records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	File's audit options changed.
NOTAUTHU	01	Not authorized to change the user audit options on the specified file.
NOTAUTHA	02	Not authorized to change the auditor audit options on the specified file.
INSSECL	03	Insufficient security label.

The change directory record extension

Table 71 on page 234 describes the format of a record that is created by changing directories.

The event qualifiers that can be associated with a directory search event are shown in Table 72 on page 236.

Table 71. Format of the change directory record extension (event code 32)					
Field name	Type	Length	Position		Comments
			Start	End	
CDIR_CLASS	Char	8	282	289	Class name.
CDIR_USER_NAME	Char	20	291	310	The name associated with the user ID.
CDIR_UTK_ENCR	Yes/No	4	312	315	Is the UTKEN associated with this user encrypted?
CDIR_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
CDIR_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
CDIR_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
CDIR_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?

Table 71. Format of the change directory record extension (event code 32) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
CDIR_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
CDIR_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
CDIR_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
CDIR_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
CDIR_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
CDIR_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
CDIR_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
CDIR_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
CDIR_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
CDIR_UTK_SECL	Char	8	386	393	The security label of the user.
CDIR_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
CDIR_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
CDIR_UTK_SNODE	Char	8	413	420	The submitting node.
CDIR_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
CDIR_UTK_SPOE	Char	8	431	438	The port of entry.
CDIR_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
CDIR_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
CDIR_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
CDIR_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
CDIR_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
CDIR_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
CDIR_AUDIT_CODE	Char	11	494	504	Audit function code.
CDIR_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
CDIR_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
CDIR_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
CDIR_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
CDIR_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
CDIR_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
CDIR_PATH_NAME	Char	1023	572	1594	The requested path name.
CDIR_FILE_ID	Char	32	1596	1627	File ID.
CDIR_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
CDIR_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
CDIR_DCE_LINK	Char	16	1651	1666	Link to connect DCE records that originate from a single DCE request.
CDIR_AUTH_TYPE	Char	13	1668	1680	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".

Table 71. Format of the change directory record extension (event code 32) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
CDIR_DFLT_PROCESS	Yes/No	4	1682	1685	Default z/OS UNIX security environment in effect.
CDIR_UTK_NETW	Char	8	1687	1694	The port of entry network name.
CDIR_X500_SUBJECT	Char	255	1696	1950	Subject's name associated with this event.
CDIR_X500_ISSUER	Char	255	1952	2206	Issuer's name associated with this event.
CDIR_SERV_POENAME	Char	64	2208	2271	SERVAUTH resource or profile name.
CDIR_CTX_USER	Char	510	2273	2782	Authenticated user name.
CDIR_CTX_REG	Char	255	2784	3038	Authenticated user registry name.
CDIR_CTX_HOST	Char	128	3040	3167	Authenticated user host name.
CDIR_CTX_MECH	Char	16	3169	3184	Authenticated user authentication mechanism object identifier (OID).
CDIR_IDID_USER	Char	985	3186	4170	Authenticated distributed user name.
CDIR_IDID_REG	Char	1021	4172	5192	Authenticated distributed user registry name.

Table 72. Event qualifiers for change directory records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	Current working directory changed. Failures are logged as directory search events.

The change file mode record extension

Table 73 on page 236 describes the format of a record that is created by changing the access mode of a file.

The event qualifiers that can be associated with changing a file mode event are shown in Table 74 on page 239.

Table 73. Format of the change file mode record extension (event code 33)					
Field name	Type	Length	Position		Comments
			Start	End	
CMOD_CLASS	Char	8	282	289	Class name.
CMOD_USER_NAME	Char	20	291	310	The name associated with the user ID.
CMOD_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
CMOD_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
CMOD_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
CMOD_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
CMOD_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
CMOD_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
CMOD_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
CMOD_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
CMOD_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
CMOD_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?

Table 73. Format of the change file mode record extension (event code 33) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
CMOD_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
CMOD_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
CMOD_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
CMOD_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
CMOD_UTK_SECL	Char	8	386	393	The security label of the user.
CMOD_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
CMOD_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
CMOD_UTK_SNODE	Char	8	413	420	The submitting node.
CMOD_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
CMOD_UTK_SPOE	Char	8	431	438	The port of entry.
CMOD_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
CMOD_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
CMOD_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
CMOD_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
CMOD_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
CMOD_APPC_LINK	Char	16	477	492	Key to link together APPC records.
CMOD_AUDIT_CODE	Char	11	494	504	Audit function code.
CMOD_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
CMOD_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
CMOD_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
CMOD_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
CMOD_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
CMOD_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
CMOD_PATH_NAME	Char	1023	572	1594	The requested path name.
CMOD_FILE_ID	Char	32	1596	1627	File ID.
CMOD_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
CMOD_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
CMOD_OLD_S_ISGID	Yes/No	4	1651	1654	Was the S_ISGID bit requested on for this file?
CMOD_OLD_S_ISUID	Yes/No	4	1656	1659	Was the S_ISUID bit requested on for this file?
CMOD_OLD_S_ISVTX	Yes/No	4	1661	1664	Was the S_ISVTX bit requested on for this file?
CMOD_OLD_OWN_READ	Yes/No	4	1666	1669	Was the owner READ bit on for this file?
CMOD_OLD_OWN_WRITE	Yes/No	4	1671	1674	Was the owner WRITE bit on for this file?
CMOD_OLD_OWN_EXEC	Yes/No	4	1676	1679	Was the owner EXECUTE bit on for this file?
CMOD_OLD_GRP_READ	Yes/No	4	1681	1684	Was the group READ bit on for this file?
CMOD_OLD_GRP_WRITE	Yes/No	4	1686	1689	Was the group WRITE bit on for this file?
CMOD_OLD_GRP_EXEC	Yes/No	4	1691	1694	Was the group EXECUTE bit on for this file?
CMOD_OLD_OTH_READ	Yes/No	4	1696	1699	Was the other READ bit on for this file?

Table 73. Format of the change file mode record extension (event code 33) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
CMOD_OLD_OTH_WRITE	Yes/No	4	1701	1704	Was the other WRITE bit on for this file?
CMOD_OLD_OTH_EXEC	Yes/No	4	1706	1709	Was the other EXECUTE bit on for this file?
CMOD_NEW_S_ISGID	Yes/No	4	1711	1714	Is the S_ISGID bit requested on for this file?
CMOD_NEW_S_ISUID	Yes/No	4	1716	1719	Is the S_ISUID bit requested on for this file?
CMOD_NEW_S_ISVTX	Yes/No	4	1721	1724	Is the S_ISVTX bit requested on for this file?
CMOD_NEW_OWN_READ	Yes/No	4	1726	1729	Is the owner READ bit on for this file?
CMOD_NEW_OWN_WRITE	Yes/No	4	1731	1734	Is the owner WRITE bit on for this file?
CMOD_NEW_OWN_EXEC	Yes/No	4	1736	1739	Is the owner EXECUTE bit on for this file?
CMOD_NEW_GRP_READ	Yes/No	4	1741	1744	Is the group READ bit on for this file?
CMOD_NEW_GRP_WRITE	Yes/No	4	1746	1749	Is the group WRITE bit on for this file?
CMOD_NEW_GRP_EXEC	Yes/No	4	1751	1754	Is the group EXECUTE bit on for this file?
CMOD_NEW_OTH_READ	Yes/No	4	1756	1759	Is the other READ bit on for this file?
CMOD_NEW_OTH_WRITE	Yes/No	4	1761	1764	Is the other WRITE bit on for this file?
CMOD_NEW_OTH_EXEC	Yes/No	4	1766	1769	Is the other EXECUTE bit on for this file?
CMOD_REQ_S_ISGID	Yes/No	4	1771	1774	Was the S_ISGID bit requested on for this file?
CMOD_REQ_S_ISUID	Yes/No	4	1776	1779	Was the S_ISUID bit requested on for this file?
CMOD_REQ_S_ISVTX	Yes/No	4	1781	1784	Was the S_ISVTX bit requested on for this file?
CMOD_REQ_OWN_READ	Yes/No	4	1786	1789	Was the owner READ bit requested on for this file?
CMOD_REQ_OWN_WRITE	Yes/No	4	1791	1794	Was the owner WRITE bit requested on for this file?
CMOD_REQ_OWN_EXEC	Yes/No	4	1796	1799	Was the owner EXECUTE bit requested on for this file?
CMOD_REQ_GRP_READ	Yes/No	4	1801	1804	Was the group READ bit requested on for this file?
CMOD_REQ_GRP_WRITE	Yes/No	4	1806	1809	Was the group WRITE bit requested on for this file?
CMOD_REQ_GRP_EXEC	Yes/No	4	1811	1814	Was the group EXECUTE bit requested on for this file?
CMOD_REQ_OTH_READ	Yes/No	4	1816	1819	Was the other READ bit requested on for this file?
CMOD_REQ_OTH_WRITE	Yes/No	4	1821	1824	Was the other WRITE bit requested on for this file?
CMOD_REQ_OTH_EXEC	Yes/No	4	1826	1829	Was the other EXECUTE bit requested on for this file?
CMOD_FILEPOOL	Char	8	1831	1838	SFS filepool containing the BFS file.
CMOD_FILESPACE	Char	8	1840	1847	SFS filespace containing the BFS file.
CMOD_INODE	Integer	10	1849	1858	Inode (file serial number).
CMOD_SCID	Integer	10	1860	1869	File SCID.
CMOD_DCE_LINK	Char	16	1871	1886	Link to connect DCE records that originate from a single DCE request.
CMOD_AUTH_TYPE	Char	13	1888	1900	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
CMOD_DFLT_PROCESS	Yes/No	4	1902	1905	Default z/OS UNIX security environment in effect.
CMOD_UTK_NETW	Char	8	1907	1914	The port of entry network name.
CMOD_X500_SUBJECT	Char	255	1916	2170	Subject's name associated with this event.
CMOD_X500_ISSUER	Char	255	2172	2426	Issuer's name associated with this event.
CMOD_SECL	Char	8	2428	2435	Security label of the resource.

Table 73. Format of the change file mode record extension (event code 33) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
CMOD_SERV_POENAME	Char	64	2437	2500	SERVAUTH resource or profile name.
CMOD_CTX_USER	Char	510	2502	3011	Authenticated user name.
CMOD_CTX_REG	Char	255	3013	3267	Authenticated user registry name.
CMOD_CTX_HOST	Char	128	3269	3396	Authenticated user host name.
CMOD_CTX_MECH	Char	16	3398	3413	Authenticated user authentication mechanism object identifier (OID).
CMOD_IDID_USER	Char	985	3415	4399	Authenticated distributed user name.
CMOD_IDID_REG	Char	1021	4401	5421	Authenticated distributed user registry name.

Table 74. Event qualifiers for change file mode records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	File's mode changed.
NOTAUTH	01	Not authorized to change the file's mode.
INSSECL	02	Insufficient security label.

The change file ownership record extension

Table 75 on page 239 describes the format of a record that is created by changing the ownership of a file.

The event qualifiers that can be associated with changing a file's ownership are shown in Table 76 on page 241.

Table 75. Format of the change file ownership record extension (event code 34)					
Field name	Type	Length	Position		Comments
			Start	End	
COWN_CLASS	Char	8	282	289	Class name.
COWN_USER_NAME	Char	20	291	310	The name associated with the user ID.
COWN_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
COWN_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
COWN_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
COWN_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
COWN_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
COWN_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
COWN_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
COWN_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
COWN_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
COWN_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
COWN_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
COWN_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
COWN_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
COWN_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?

Table 75. Format of the change file ownership record extension (event code 34) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
COWN_UTK_SECL	Char	8	386	393	The security label of the user.
COWN_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
COWN_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
COWN_UTK_SNODE	Char	8	413	420	The submitting node.
COWN_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
COWN_UTK_SPOE	Char	8	431	438	The port of entry.
COWN_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
COWN_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
COWN_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
COWN_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
COWN_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
COWN_APPC_LINK	Char	16	477	492	Key to link together APPC records.
COWN_AUDIT_CODE	Char	11	494	504	Audit function code.
COWN_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
COWN_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
COWN_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
COWN_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
COWN_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
COWN_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
COWN_PATH_NAME	Char	1023	572	1594	The requested path name.
COWN_FILE_ID	Char	32	1596	1627	File ID.
COWN_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
COWN_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
COWN_UID	Integer	10	1651	1660	The z/OS UNIX user identifier (UID) input parameter.
COWN_GID	Integer	10	1662	1671	The z/OS UNIX group identifier (GID) input parameter.
COWN_FILEPOOL	Char	8	1673	1680	SFS filepool containing the BFS file.
COWN_FILESPACE	Char	8	1682	1689	SFS filespace containing the BFS file.
COWN_INODE	Integer	10	1691	1700	Inode (file serial number).
COWN_SCID	Integer	10	1702	1711	File SCID.
COWN_DCE_LINK	Char	16	1713	1728	Link to connect DCE records that originate from a single DCE request.
COWN_AUTH_TYPE	Char	13	1730	1742	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
COWN_DFLT_PROCESS	Yes/No	4	1744	1747	Default z/OS UNIX security environment in effect.
COWN_UTK_NETW	Char	8	1749	1756	The port of entry network name.
COWN_X500_SUBJECT	Char	255	1758	2012	Subject's name associated with this event.
COWN_X500_ISSUER	Char	255	2014	2268	Issuer's name associated with this event.
COWN_SECL	Char	8	2270	2277	Security label of the resource.

Table 75. Format of the change file ownership record extension (event code 34) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
COWN_SERV_POENAME	Char	64	2279	2342	SERVAUTH resource or profile name.
COWN_CTX_USER	Char	510	2344	2853	Authenticated user name.
COWN_CTX_REG	Char	255	2855	3109	Authenticated user registry name.
COWN_CTX_HOST	Char	128	3111	3238	Authenticated user host name.
COWN_CTX_MECH	Char	16	3240	3255	Authenticated user authentication mechanism object identifier (OID).
COWN_IDID_USER	Char	985	3257	4241	Authenticated distributed user name.
COWN_IDID_REG	Char	1021	4243	5263	Authenticated distributed user registry name.

Table 76. Event qualifiers for change file ownership records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	File's ownership changed.
NOTAUTH	01	Not authorized to change the file's ownership.
INSSECL	02	Insufficient security label.

The clear SETID bits record extension

Table 77 on page 241 describes the format of a record that is created by clearing the SETID bits of a file.

The event qualifier that can be associated with clearing a file's SETID bits is shown in Table 78 on page 243.

Table 77. Format of the clear SETID bits record extension (event code 35)					
Field name	Type	Length	Position		Comments
			Start	End	
CSID_CLASS	Char	8	282	289	Class name.
CSID_USER_NAME	Char	20	291	310	The name associated with the user ID.
CSID_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
CSID_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
CSID_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
CSID_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
CSID_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
CSID_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
CSID_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
CSID_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
CSID_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
CSID_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
CSID_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
CSID_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
CSID_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
CSID_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?

Table 77. Format of the clear SETID bits record extension (event code 35) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
CSID_UTK_SECL	Char	8	386	393	The security label of the user.
CSID_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
CSID_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
CSID_UTK_SNODE	Char	8	413	420	The submitting node.
CSID_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
CSID_UTK_SPOE	Char	8	431	438	The port of entry.
CSID_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
CSID_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
CSID_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
CSID_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
CSID_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
CSID_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
CSID_AUDIT_CODE	Char	11	494	504	Audit function code.
CSID_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
CSID_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
CSID_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
CSID_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
CSID_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
CSID_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
CSID_PATH_NAME	Char	1023	572	1594	The requested path name.
CSID_FILE_ID	Char	32	1596	1627	File ID.
CSID_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
CSID_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
CSID_OLD_S_ISGID	Yes/No	4	1651	1654	Was the S_ISGID bit requested on for this file?
CSID_OLD_S_ISUID	Yes/No	4	1656	1659	Was the S_ISUID bit requested on for this file?
CSID_OLD_S_ISVTX	Yes/No	4	1661	1664	Was the S_ISVTX bit requested on for this file?
CSID_OLD_OWN_READ	Yes/No	4	1666	1669	Was the owner READ bit on for this file?
CSID_OLD_OWN_WRITE	Yes/No	4	1671	1674	Was the owner WRITE bit on for this file?
CSID_OLD_OWN_EXEC	Yes/No	4	1676	1679	Was the owner EXECUTE bit on for this file?
CSID_OLD_GRP_READ	Yes/No	4	1681	1684	Was the group READ bit on for this file?
CSID_OLD_GRP_WRITE	Yes/No	4	1686	1689	Was the group WRITE bit on for this file?
CSID_OLD_GRP_EXEC	Yes/No	4	1691	1694	Was the group EXECUTE bit on for this file?
CSID_OLD_OTH_READ	Yes/No	4	1696	1699	Was the other READ bit on for this file?
CSID_OLD_OTH_WRITE	Yes/No	4	1701	1704	Was the other WRITE bit on for this file?
CSID_OLD_OTH_EXEC	Yes/No	4	1706	1709	Was the other EXECUTE bit on for this file?
CSID_NEW_S_ISGID	Yes/No	4	1711	1714	Is the S_ISGID bit requested on for this file?

Table 77. Format of the clear SETID bits record extension (event code 35) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
CSID_NEW_S_ISUID	Yes/No	4	1716	1719	Is the S_ISUID bit requested on for this file?
CSID_NEW_S_ISVTX	Yes/No	4	1721	1724	Is the S_ISVTX bit requested on for this file?
CSID_NEW_OWN_READ	Yes/No	4	1726	1729	Is the owner READ bit on for this file?
CSID_NEW_OWN_WRITE	Yes/No	4	1731	1734	Is the owner WRITE bit on for this file?
CSID_NEW_OWN_EXEC	Yes/No	4	1736	1739	Is the owner EXECUTE bit on for this file?
CSID_NEW_GRP_READ	Yes/No	4	1741	1744	Is the group READ bit on for this file?
CSID_NEW_GRP_WRITE	Yes/No	4	1746	1749	Is the group WRITE bit on for this file?
CSID_NEW_GRP_EXEC	Yes/No	4	1751	1754	Is the group EXECUTE bit on for this file?
CSID_NEW_OTH_READ	Yes/No	4	1756	1759	Is the other READ bit on for this file?
CSID_NEW_OTH_WRITE	Yes/No	4	1761	1764	Is the other WRITE bit on for this file?
CSID_NEW_OTH_EXEC	Yes/No	4	1766	1769	Is the other EXECUTE bit on for this file?
CSID_DFLT_PROCESS	Yes/No	4	1771	1774	Default z/OS UNIX security environment in effect.
CSID_UTK_NETW	Char	8	1776	1783	The port of entry network name.
CSID_X500_SUBJECT	Char	255	1785	2039	Subject's name associated with this event.
CSID_X500_ISSUER	Char	255	2041	2295	Issuer's name associated with this event.
CSID_SERV_POENAME	Char	64	2297	2360	SERVAUTH resource or profile name.
CSID_CTX_USER	Char	510	2362	2871	Authenticated user name.
CSID_CTX_REG	Char	255	2873	3127	Authenticated user registry name.
CSID_CTX_HOST	Char	128	3129	3256	Authenticated user host name.
CSID_CTX_MECH	Char	16	3258	3273	Authenticated user authentication mechanism object identifier (OID).
CSID_IDID_USER	Char	985	3275	4259	Authenticated distributed user name.
CSID_IDID_REG	Char	1021	4261	5281	Authenticated distributed user registry name.

Table 78. Event qualifiers for clear SETID records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	S_ISUID, S_ISGID, and S_ISVTX changed. There are no failure cases for this event.

The EXEC SETUID/SETGID record extension

Table 79 on page 243 describes the format of a record that is created by the execution of an EXEC SETUID or SETGID.

The event qualifier that can be associated with the execution of EXEC SETUID or EXEC SETGID is shown in Table 80 on page 245.

Table 79. Format of the EXEC with SETUID/SETGID record extension (event code 36)					
Field name	Type	Length	Position		Comments
			Start	End	
ESID_CLASS	Char	8	282	289	Class name.
ESID_USER_NAME	Char	20	291	310	The name associated with the user ID.

Table 79. Format of the EXEC with SETUID/SETGID record extension (event code 36) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
ESID_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
ESID_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
ESID_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
ESID_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
ESID_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
ESID_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
ESID_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
ESID_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
ESID_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
ESID_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
ESID_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
ESID_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
ESID_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
ESID_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
ESID_UTK_SECL	Char	8	386	393	The security label of the user.
ESID_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
ESID_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
ESID_UTK_SNODE	Char	8	413	420	The submitting node.
ESID_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
ESID_UTK_SPOE	Char	8	431	438	The port of entry.
ESID_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ESID_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
ESID_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
ESID_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
ESID_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
ESID_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
ESID_AUDIT_CODE	Char	11	494	504	Audit function code.
ESID_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
ESID_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
ESID_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
ESID_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
ESID_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
ESID_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
ESID_NEW_REAL_UID	Integer	10	572	581	New real z/OS UNIX user identifier (UID).
ESID_NEW_EFF_UID	Integer	10	583	592	New effective z/OS UNIX user identifier (UID).
ESID_NEW_SAVED_UID	Integer	10	594	603	New saved z/OS UNIX user identifier (UID).
ESID_NEW_REAL_GID	Integer	10	605	614	New real z/OS UNIX group identifier (GID).

Table 79. Format of the EXEC with SETUID/SETGID record extension (event code 36) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
ESID_NEW_EFF_GID	Integer	10	616	625	New effective z/OS UNIX group identifier (GID).
ESID_NEW_SAVED_GID	Integer	10	627	636	New saved z/OS UNIX group identifier (GID).
ESID_UID	Integer	10	638	647	The z/OS UNIX user identifier (UID) input parameter.
ESID_GID	Integer	10	649	658	The z/OS UNIX group identifier (GID) input parameter.
ESID_DFLT_PROCESS	Yes/No	4	660	663	Default z/OS UNIX security environment in effect.
ESID_UTK_NETW	Char	8	665	672	The port of entry network name.
ESID_X500_SUBJECT	Char	255	674	928	Subject's name associated with this event.
ESID_X500_ISSUER	Char	255	930	1184	Issuer's name associated with this event.
ESID_SERV_POENAME	Char	64	1186	1249	SERVAUTH resource or profile name.
ESID_CTX_USER	Char	510	1251	1760	Authenticated user name.
ESID_CTX_REG	Char	255	1762	2016	Authenticated user registry name.
ESID_CTX_HOST	Char	128	2018	2145	Authenticated user host name.
ESID_CTX_MECH	Char	16	2147	2162	Authenticated user authentication mechanism object identifier (OID).
ESID_IDID_USER	Char	985	2164	3148	Authenticated distributed user name.
ESID_IDID_REG	Char	1021	3150	4170	Authenticated distributed user registry name.

Table 80. Event qualifiers for EXEC with SETID/SETGID records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	z/OS UNIX user identifier (UID) or z/OS UNIX group identifier (GID) changed. There are no failure cases for this event.

The GETPSENT record extension

Table 81 on page 245 describes the format of a record that is created by the GETPSENT service.

The event qualifiers that can be associated with the GETPSENT service are shown in [Table 82 on page 247](#).

Table 81. Format of the GETPSENT record extension (event code 37)					
Field name	Type	Length	Position		Comments
			Start	End	
GPST_CLASS	Char	8	282	289	Class name.
GPST_USER_NAME	Char	20	291	310	The name associated with the user ID.
GPST_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
GPST_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
GPST_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
GPST_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
GPST_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
GPST_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
GPST_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
GPST_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?

Table 81. Format of the GETPSENT record extension (event code 37) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
GPST_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
GPST_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
GPST_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
GPST_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
GPST_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
GPST_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
GPST_UTK_SECL	Char	8	386	393	The security label of the user.
GPST_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
GPST_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
GPST_UTK_SNODE	Char	8	413	420	The submitting node.
GPST_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
GPST_UTK_SPOE	Char	8	431	438	The port of entry.
GPST_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
GPST_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
GPST_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
GPST_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
GPST_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
GPST_APPC_LINK	Char	16	477	492	Key to link together APPC records.
GPST_AUDIT_CODE	Char	11	494	504	Audit function code.
GPST_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
GPST_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
GPST_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
GPST_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
GPST_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
GPST_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
GPST_TGT_REAL_UID	Integer	10	572	581	Target real z/OS UNIX user identifier (UID).
GPST_TGT_EFF_UID	Integer	10	583	592	Target effective z/OS UNIX user identifier (UID).
GPST_TGT_SAV_UID	Integer	10	594	603	Target saved z/OS UNIX user identifier (UID).
GPST_TGT_PID	Integer	10	605	614	Target process ID.
GPST_DFLT_PROCESS	Yes/No	4	616	619	Default z/OS UNIX security environment in effect.
GPST_UTK_NETW	Char	8	621	628	The port of entry network name.
GPST_X500_SUBJECT	Char	255	630	884	Subject's name associated with this event.
GPST_X500_ISSUER	Char	255	886	1140	Issuer's name associated with this event.
GPST_SERV_POENAME	Char	64	1142	1205	SERVAUTH resource or profile name.
GPST_CTX_USER	Char	510	1207	1716	Authenticated user name.
GPST_CTX_REG	Char	255	1718	1972	Authenticated user registry name.
GPST_CTX_HOST	Char	128	1974	2101	Authenticated user host name.

Table 81. Format of the GETPSENT record extension (event code 37) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
GPST_CTX_MECH	Char	16	2103	2118	Authenticated user authentication mechanism object identifier (OID).
GPST_IDID_USER	Char	985	2120	3104	Authenticated distributed user name.
GPST_IDID_REG	Char	1021	3106	4126	Authenticated distributed user registry name.

Table 82. Event qualifiers for GETPSENT records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	GETPSENT was successful.
NOTAUTH	01	Not authorized to the specified process.

The initialize z/OS UNIX record extension

Table 83 on page 247 describes the format of a record that is created when a z/OS UNIX process is initialized.

The event qualifiers that can be associated with the initiation of a z/OS UNIX process are shown in Table 84 on page 248.

Table 83. Format of the initialize z/OS UNIX process record extension (event code 38)					
Field name	Type	Length	Position		Comments
			Start	End	
IOEP_CLASS	Char	8	282	289	Class name.
IOEP_USER_NAME	Char	20	291	310	The name associated with the user ID.
IOEP_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
IOEP_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
IOEP_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
IOEP_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
IOEP_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
IOEP_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
IOEP_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
IOEP_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
IOEP_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
IOEP_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
IOEP_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
IOEP_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
IOEP_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
IOEP_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
IOEP_UTK_SECL	Char	8	386	393	The security label of the user.
IOEP_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
IOEP_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
IOEP_UTK_SNODE	Char	8	413	420	The submitting node.

Table 83. Format of the initialize z/OS UNIX process record extension (event code 38) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
IOEP_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
IOEP_UTK_SPOE	Char	8	431	438	The port of entry.
IOEP_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
IOEP_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
IOEP_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
IOEP_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
IOEP_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
IOEP_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
IOEP_AUDIT_CODE	Char	11	494	504	Audit function code.
IOEP_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
IOEP_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
IOEP_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
IOEP_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
IOEP_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
IOEP_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
IOEP_DFLT_PROCESS	Yes/No	4	572	575	Default z/OS UNIX security environment in effect.
IOEP_UTK_NETW	Char	8	577	584	The port of entry network name.
IOEP_X500_SUBJECT	Char	255	586	840	Subject's name associated with this event.
IOEP_X500_ISSUER	Char	255	842	1096	Issuer's name associated with this event.
IOEP_SERV_POENAME	Char	64	1098	1161	SERVAUTH resource or profile name.
IOEP_CTX_USER	Char	510	1163	1672	Authenticated user name.
IOEP_CTX_REG	Char	255	1674	1928	Authenticated user registry name.
IOEP_CTX_HOST	Char	128	1930	2057	Authenticated user host name.
IOEP_CTX_MECH	Char	16	2059	2074	Authenticated user authentication mechanism object identifier (OID).
IOEP_IDID_USER	Char	985	2076	3060	Authenticated distributed user name.
IOEP_IDID_REG	Char	1021	3062	4082	Authenticated distributed user registry name.

Table 84. Event qualifiers for initialize z/OS UNIX process records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Process successfully initialized.
NOTDFND	01	User not defined as a z/OS UNIX user. The OMVS segment or the user profile was missing.
NOUID	02	Incompletely defined user ID. There was no z/OS UNIX user identifier (UID) in profile.
NOGID	03	User's current group has no z/OS UNIX group identifier (GID).

The z/OS UNIX process completion record

Table 85 on page 249 describes the format of a record that is created when a z/OS UNIX process completes.

The event qualifier that can be associated with the completion of a z/OS UNIX process is shown in Table 86 on page 250.

Table 85. Format of the z/OS UNIX process complete record extension (event code 39)					
Field name	Type	Length	Position		Comments
			Start	End	
TOEP_CLASS	Char	8	282	289	Class name.
TOEP_USER_NAME	Char	20	291	310	The name associated with the user ID.
TOEP_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
TOEP_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
TOEP_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
TOEP_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
TOEP_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
TOEP_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
TOEP_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
TOEP_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
TOEP_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
TOEP_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
TOEP_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
TOEP_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
TOEP_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
TOEP_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
TOEP_UTK_SECL	Char	8	386	393	The security label of the user.
TOEP_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
TOEP_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
TOEP_UTK_SNODE	Char	8	413	420	The submitting node.
TOEP_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
TOEP_UTK_SPOE	Char	8	431	438	The port of entry.
TOEP_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
TOEP_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
TOEP_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
TOEP_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
TOEP_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
TOEP_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
TOEP_AUDIT_CODE	Char	11	494	504	Audit function code.
TOEP_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
TOEP_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
TOEP_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).

Table 85. Format of the z/OS UNIX process complete record extension (event code 39) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
TOEP_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
TOEP_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
TOEP_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
TOEP_DFLT_PROCESS	Yes/No	4	572	575	Default z/OS UNIX security environment in effect.
TOEP_UTK_NETW	Char	8	577	584	The port of entry network name.
TOEP_X500_SUBJECT	Char	255	586	840	Subject's name associated with this event.
TOEP_X500_ISSUER	Char	255	842	1096	Issuer's name associated with this event.
TOEP_SERV_POENAME	Char	64	1098	1161	SERVAUTH resource or profile name.
TOEP_CTX_USER	Char	510	1163	1672	Authenticated user name.
TOEP_CTX_REG	Char	255	1674	1928	Authenticated user registry name.
TOEP_CTX_HOST	Char	128	1930	2057	Authenticated user host name.
TOEP_CTX_MECH	Char	16	2059	2074	Authenticated user authentication mechanism object identifier (OID).
TOEP_IDID_USER	Char	985	2076	3060	Authenticated distributed user name.
TOEP_IDID_REG	Char	1021	3062	4082	Authenticated distributed user registry name.

Table 86. Event qualifiers for z/OS UNIX process complete records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	Process complete. There are no failure cases for this event.

The KILL record extension

Table 87 on page 250 describes the format of a record that is created by the termination with extreme prejudice of a process.

The event qualifiers that can be associated with the killing of a process are shown in Table 88 on page 252.

Table 87. Format of the KILL process record extension (event code 40)					
Field name	Type	Length	Position		Comments
			Start	End	
KILL_CLASS	Char	8	282	289	Class name.
KILL_USER_NAME	Char	20	291	310	The name associated with the user ID.
KILL_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
KILL_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
KILL_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
KILL_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
KILL_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
KILL_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
KILL_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
KILL_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?

Table 87. Format of the KILL process record extension (event code 40) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
KILL_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
KILL_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
KILL_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
KILL_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
KILL_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
KILL_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
KILL_UTK_SECL	Char	8	386	393	The security label of the user.
KILL_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
KILL_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
KILL_UTK_SNODE	Char	8	413	420	The submitting node.
KILL_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
KILL_UTK_SPOE	Char	8	431	438	The port of entry.
KILL_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
KILL_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
KILL_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
KILL_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
KILL_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
KILL_APPC_LINK	Char	16	477	492	Key to link together APPC records.
KILL_AUDIT_CODE	Char	11	494	504	Audit function code.
KILL_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
KILL_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
KILL_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
KILL_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
KILL_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
KILL_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
KILL_TGT_REAL_UID	Integer	10	572	581	Target real z/OS UNIX user identifier (UID).
KILL_TGT_EFF_UID	Integer	10	583	592	Target effective z/OS UNIX user identifier (UID).
KILL_TGT_SAV_UID	Integer	10	594	603	Target saved z/OS UNIX user identifier (UID).
KILL_TGT_PID	Integer	10	605	614	Target process ID.
KILL_SIGNAL_CODE	Integer	10	616	625	Kill signal code.
KILL_DFLT_PROCESS	Yes/No	4	627	630	Default z/OS UNIX security environment in effect.
KILL_UTK_NETW	Char	8	632	639	The port of entry network name.
KILL_X500_SUBJECT	Char	255	641	895	Subject's name associated with this event.
KILL_X500_ISSUER	Char	255	897	1151	Issuer's name associated with this event.
KILL_SECL	Char	8	1153	1160	Security label of the resource.
KILL_SERV_POENAME	Char	64	1162	1225	SERVAUTH resource or profile name.
KILL_CTX_USER	Char	510	1227	1736	Authenticated user name.

Table 87. Format of the KILL process record extension (event code 40) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
KILL_CTX_REG	Char	255	1738	1992	Authenticated user registry name.
KILL_CTX_HOST	Char	128	1994	2121	Authenticated user host name.
KILL_CTX_MECH	Char	16	2123	2138	Authenticated user authentication mechanism object identifier (OID).
KILL_IDID_USER	Char	985	2140	3124	Authenticated distributed user name.
KILL_IDID_REG	Char	1021	3126	4146	Authenticated distributed user registry name.

Table 88. Event qualifiers for KILL records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Process terminated.
NOTAUTH	01	Not authorized to kill the specified process.
INSSECL	02	Insufficient security label.

The LINK record extension

Table 89 on page 252 describes the format of a record that is created by a LINK operation.

The event qualifier that can be associated with a LINK event is shown in Table 90 on page 254.

Table 89. Format of the LINK record extension (event code 41)					
Field name	Type	Length	Position		Comments
			Start	End	
LINK_CLASS	Char	8	282	289	Class name.
LINK_USER_NAME	Char	20	291	310	The name associated with the user ID.
LINK_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
LINK_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
LINK_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
LINK_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
LINK_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
LINK_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
LINK_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
LINK_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
LINK_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
LINK_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
LINK_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
LINK_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
LINK_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
LINK_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
LINK_UTK_SECL	Char	8	386	393	The security label of the user.
LINK_UTK_EXECNODE	Char	8	395	402	The execution node of the work.

Table 89. Format of the LINK record extension (event code 41) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
LINK_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
LINK_UTK_SNODE	Char	8	413	420	The submitting node.
LINK_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
LINK_UTK_SPOE	Char	8	431	438	The port of entry.
LINK_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
LINK_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
LINK_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
LINK_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
LINK_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
LINK_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
LINK_AUDIT_CODE	Char	11	494	504	Audit function code.
LINK_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
LINK_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
LINK_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
LINK_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
LINK_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
LINK_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
LINK_PATH_NAME	Char	1023	572	1594	The requested path name.
LINK_FILE_ID	Char	32	1596	1627	File ID.
LINK_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
LINK_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
LINK_REQUEST_PATH2	Char	1023	1651	2673	Second requested path name.
LINK_PATH_TYPE	Char	4	2675	2678	Type of the requested path name. Valid values are "OLD" and "NEW".
LINK_FILEPOOL	Char	8	2680	2687	SFS filepool containing the BFS file.
LINK_FILESPACE	Char	8	2689	2696	SFS filespace containing the BFS filespace.
LINK_INODE	Integer	10	2698	2707	Inode (file serial number).
LINK_SCID	Integer	10	2709	2718	File SCID.
LINK_DCE_LINK	Char	16	2720	2735	Link to connect DCE records that originate from a single DCE request.
LINK_AUTH_TYPE	Char	13	2737	2749	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
LINK_DFLT_PROCESS	Yes/No	4	2751	2754	Default z/OS UNIX security environment in effect.
LINK_UTK_NETW	Char	8	2756	2763	The port of entry network name.
LINK_X500_SUBJECT	Char	255	2765	3019	Subject's name associated with this event.
LINK_X500_ISSUER	Char	255	3021	3275	Issuer's name associated with this event.
LINK_SERV_POENAME	Char	64	3277	3340	SERVAUTH resource or profile name.

Table 89. Format of the LINK record extension (event code 41) (continued)

Field name	Type	Length	Position		Comments
			Start	End	
LINK_CTX_USER	Char	510	3342	3851	Authenticated user name.
LINK_CTX_REG	Char	255	3853	4107	Authenticated user registry name.
LINK_CTX_HOST	Char	128	4109	4236	Authenticated user host name.
LINK_CTX_MECH	Char	16	4238	4253	Authenticated user authentication mechanism object identifier (OID).
LINK_IDID_USER	Char	985	4255	5239	Authenticated distributed user name.
LINK_IDID_REG	Char	1021	5241	6261	Authenticated distributed user registry name.

Table 90. Event qualifiers for LINK records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	New link created. There are no failure cases for this event.

The MKDIR record extension

Table 91 on page 254 describes the format of a record that is created by making a directory.

The event qualifier that can be associated with making a directory is shown in Table 92 on page 257.

Table 91. Format of the MKDIR record extension (event code 42)

Field name	Type	Length	Position		Comments
			Start	End	
MDIR_CLASS	Char	8	282	289	Class name.
MDIR_USER_NAME	Char	20	291	310	The name associated with the user ID.
MDIR_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
MDIR_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
MDIR_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
MDIR_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
MDIR_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
MDIR_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
MDIR_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
MDIR_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
MDIR_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
MDIR_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
MDIR_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
MDIR_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
MDIR_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
MDIR_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
MDIR_UTK_SECL	Char	8	386	393	The security label of the user.
MDIR_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
MDIR_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.

Table 91. Format of the MKDIR record extension (event code 42) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
MDIR_UTK_SNODE	Char	8	413	420	The submitting node.
MDIR_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
MDIR_UTK_SPOE	Char	8	431	438	The port of entry.
MDIR_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
MDIR_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
MDIR_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
MDIR_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
MDIR_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
MDIR_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
MDIR_AUDIT_CODE	Char	11	494	504	Audit function code.
MDIR_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
MDIR_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
MDIR_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
MDIR_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
MDIR_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
MDIR_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
MDIR_PATH_NAME	Char	1023	572	1594	The requested path name.
MDIR_FILE_ID	Char	32	1596	1627	File ID.
MDIR_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
MDIR_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
MDIR_OLD_S_ISGID	Yes/No	4	1651	1654	Was the S_ISGID bit requested on for this file?
MDIR_OLD_S_ISUID	Yes/No	4	1656	1659	Was the S_ISUID bit requested on for this file?
MDIR_OLD_S_ISVTX	Yes/No	4	1661	1664	Was the S_ISVTX bit requested on for this file?
MDIR_OLD_OWN_READ	Yes/No	4	1666	1669	Was the owner READ bit on for this file?
MDIR_OLD_OWN_WRITE	Yes/No	4	1671	1674	Was the owner WRITE bit on for this file?
MDIR_OLD_OWN_EXEC	Yes/No	4	1676	1679	Was the owner EXECUTE bit on for this file?
MDIR_OLD_GRP_READ	Yes/No	4	1681	1684	Was the group READ bit on for this file?
MDIR_OLD_GRP_WRITE	Yes/No	4	1686	1689	Was the group WRITE bit on for this file?
MDIR_OLD_GRP_EXEC	Yes/No	4	1691	1694	Was the group EXECUTE bit on for this file?
MDIR_OLD_OTH_READ	Yes/No	4	1696	1699	Was the other READ bit on for this file?
MDIR_OLD_OTH_WRITE	Yes/No	4	1701	1704	Was the other WRITE bit on for this file?
MDIR_OLD_OTH_EXEC	Yes/No	4	1706	1709	Was the other EXECUTE bit on for this file?
MDIR_NEW_S_ISGID	Yes/No	4	1711	1714	Is the S_ISGID bit requested on for this file?
MDIR_NEW_S_ISUID	Yes/No	4	1716	1719	Is the S_ISUID bit requested on for this file?
MDIR_NEW_S_ISVTX	Yes/No	4	1721	1724	Is the S_ISVTX bit requested on for this file?
MDIR_NEW_OWN_READ	Yes/No	4	1726	1729	Is the owner READ bit on for this file?

Table 91. Format of the MKDIR record extension (event code 42) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
MDIR_NEW_OWN_WRITE	Yes/No	4	1731	1734	Is the owner WRITE bit on for this file?
MDIR_NEW_OWN_EXEC	Yes/No	4	1736	1739	Is the owner EXECUTE bit on for this file?
MDIR_NEW_GRP_READ	Yes/No	4	1741	1744	Is the group READ bit on for this file?
MDIR_NEW_GRP_WRITE	Yes/No	4	1746	1749	Is the group WRITE bit on for this file?
MDIR_NEW_GRP_EXEC	Yes/No	4	1751	1754	Is the group EXECUTE bit on for this file?
MDIR_NEW_OTH_READ	Yes/No	4	1756	1759	Is the other READ bit on for this file?
MDIR_NEW_OTH_WRITE	Yes/No	4	1761	1764	Is the other WRITE bit on for this file?
MDIR_NEW_OTH_EXEC	Yes/No	4	1766	1769	Is the other EXECUTE bit on for this file?
MDIR_UNEW_READ	Char	8	1771	1778	What are the new user audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MDIR_UNEW_WRITE	Char	8	1780	1787	What are the new user audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MDIR_UNEW_EXEC	Char	8	1789	1796	What are the new user audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MDIR_ANEW_READ	Char	8	1798	1805	What are the new auditor audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MDIR_ANEW_WRITE	Char	8	1807	1814	What are the new auditor audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MDIR_ANEW_EXEC	Char	8	1816	1823	What are the new auditor audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MDIR_REQ_S_ISGID	Yes/No	4	1825	1828	Was the S_ISGID bit requested on for this file?
MDIR_REQ_S_ISUID	Yes/No	4	1830	1833	Was the S_ISUID bit requested on for this file?
MDIR_REQ_S_ISVTX	Yes/No	4	1835	1838	Was the S_ISVTX bit requested on for this file?
MDIR_REQ_OWN_READ	Yes/No	4	1840	1843	Was the owner READ bit requested on for this file?
MDIR_REQ_OWN_WRITE	Yes/No	4	1845	1848	Was the owner WRITE bit requested on for this file?
MDIR_REQ_OWN_EXEC	Yes/No	4	1850	1853	Was the owner EXECUTE bit requested on for this file?
MDIR_REQ_GRP_READ	Yes/No	4	1855	1858	Was the group READ bit requested on for this file?
MDIR_REQ_GRP_WRITE	Yes/No	4	1860	1863	Was the group WRITE bit requested on for this file?
MDIR_REQ_GRP_EXEC	Yes/No	4	1865	1868	Was the group EXECUTE bit requested on for this file?
MDIR_REQ_OTH_READ	Yes/No	4	1870	1873	Was the other READ bit requested on for this file?
MDIR_REQ_OTH_WRITE	Yes/No	4	1875	1878	Was the other WRITE bit requested on for this file?
MDIR_REQ_OTH_EXEC	Yes/No	4	1880	1883	Was the other EXECUTE bit requested on for this file?
MDIR_FILEPOOL	Char	8	1885	1892	SFS filepool containing the BFS file.
MDIR_FILESPACE	Char	8	1894	1901	SFS filespace containing the BFS file.
MDIR_INODE	Integer	10	1903	1912	Inode (file serial number).
MDIR_SCID	Integer	10	1914	1923	File SCID.
MDIR_DFLT_PROCESS	Yes/No	4	1925	1928	Default z/OS UNIX security environment in effect.

Table 91. Format of the MKDIR record extension (event code 42) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
MDIR_UTK_NETW	Char	8	1930	1937	The port of entry network name.
MDIR_X500_SUBJECT	Char	255	1939	2193	Subject's name associated with this event.
MDIR_X500_ISSUER	Char	255	2195	2449	Issuer's name associated with this event.
MDIR_SECL	Char	8	2451	2458	Security label of the resource.
MDIR_SERV_POENAME	Char	64	2460	2523	SERVAUTH resource or profile name.
MDIR_CTX_USER	Char	510	2525	3034	Authenticated user name.
MDIR_CTX_REG	Char	255	3036	3290	Authenticated user registry name.
MDIR_CTX_HOST	Char	128	3292	3419	Authenticated user host name.
MDIR_CTX_MECH	Char	16	3421	3436	Authenticated user authentication mechanism object identifier (OID).
MDIR_IDID_USER	Char	985	3438	4422	Authenticated distributed user name.
MDIR_IDID_REG	Char	1021	4424	5444	Authenticated distributed user registry name.

Table 92. Event qualifiers for MKDIR records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	Directory created. There are no failure cases for this event.

The MKNOD record extension

Table 93 on page 257 describes the format of a record that is created by making a node.

The event qualifier that can be associated with making a node is shown in Table 94 on page 260.

The event qualifier that can be associated with the mounting of a file system event is shown in Table 96 on page 262.

Table 93. Format of the MKNOD record extension (event code 43)					
Field name	Type	Length	Position		Comments
			Start	End	
MNOD_CLASS	Char	8	282	289	Class name.
MNOD_USER_NAME	Char	20	291	310	The name associated with the user ID.
MNOD_UTK_ENCR	Yes/No	4	312	315	Is the UTKEN associated with this user encrypted?
MNOD_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
MNOD_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
MNOD_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
MNOD_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
MNOD_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
MNOD_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
MNOD_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
MNOD_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
MNOD_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?

Table 93. Format of the MKNOD record extension (event code 43) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
MNOD_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
MNOD_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
MNOD_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
MNOD_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
MNOD_UTK_SECL	Char	8	386	393	The security label of the user.
MNOD_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
MNOD_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
MNOD_UTK_SNODE	Char	8	413	420	The submitting node.
MNOD_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
MNOD_UTK_SPOE	Char	8	431	438	The port of entry.
MNOD_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
MNOD_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
MNOD_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
MNOD_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
MNOD_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
MNOD_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
MNOD_AUDIT_CODE	Char	11	494	504	Audit function code.
MNOD_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
MNOD_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
MNOD_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
MNOD_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
MNOD_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
MNOD_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
MNOD_PATH_NAME	Char	1023	572	1594	The requested path name.
MNOD_FILE_ID	Char	32	1596	1627	File ID.
MNOD_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
MNOD_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
MNOD_OLD_S_ISGID	Yes/No	4	1651	1654	Was the S_ISGID bit requested on for this file?
MNOD_OLD_S_ISUID	Yes/No	4	1656	1659	Was the S_ISUID bit requested on for this file?
MNOD_OLD_S_ISVTX	Yes/No	4	1661	1664	Was the S_ISVTX bit requested on for this file?
MNOD_OLD_OWN_READ	Yes/No	4	1666	1669	Was the owner READ bit on for this file?
MNOD_OLD_OWN_WRITE	Yes/No	4	1671	1674	Was the owner WRITE bit on for this file?
MNOD_OLD_OWN_EXEC	Yes/No	4	1676	1679	Was the owner EXECUTE bit on for this file?
MNOD_OLD_GRP_READ	Yes/No	4	1681	1684	Was the group READ bit on for this file?
MNOD_OLD_GRP_WRITE	Yes/No	4	1686	1689	Was the group WRITE bit on for this file?
MNOD_OLD_GRP_EXEC	Yes/No	4	1691	1694	Was the group EXECUTE bit on for this file?

Table 93. Format of the MKNOD record extension (event code 43) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
MNOD_OLD_OTH_READ	Yes/No	4	1696	1699	Was the other READ bit on for this file?
MNOD_OLD_OTH_WRITE	Yes/No	4	1701	1704	Was the other WRITE bit on for this file?
MNOD_OLD_OTH_EXEC	Yes/No	4	1706	1709	Was the other EXECUTE bit on for this file?
MNOD_NEW_S_ISGID	Yes/No	4	1711	1714	Is the S_ISGID bit requested on for this file?
MNOD_NEW_S_ISUID	Yes/No	4	1716	1719	Is the S_ISUID bit requested on for this file?
MNOD_NEW_S_ISVTX	Yes/No	4	1721	1724	Is the S_ISVTX bit requested on for this file?
MNOD_NEW_OWN_READ	Yes/No	4	1726	1729	Is the owner READ bit on for this file?
MNOD_NEW_OWN_WRITE	Yes/No	4	1731	1734	Is the owner WRITE bit on for this file?
MNOD_NEW_OWN_EXEC	Yes/No	4	1736	1739	Is the owner EXECUTE bit on for this file?
MNOD_NEW_GRP_READ	Yes/No	4	1741	1744	Is the group READ bit on for this file?
MNOD_NEW_GRP_WRITE	Yes/No	4	1746	1749	Is the group WRITE bit on for this file?
MNOD_NEW_GRP_EXEC	Yes/No	4	1751	1754	Is the group EXECUTE bit on for this file?
MNOD_NEW_OTH_READ	Yes/No	4	1756	1759	Is the other READ bit on for this file?
MNOD_NEW_OTH_WRITE	Yes/No	4	1761	1764	Is the other WRITE bit on for this file?
MNOD_NEW_OTH_EXEC	Yes/No	4	1766	1769	Is the other EXECUTE bit on for this file?
MNOD_UNEW_READ	Char	8	1771	1778	What are the new user audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MNOD_UNEW_WRITE	Char	8	1780	1787	What are the new user audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MNOD_UNEW_EXEC	Char	8	1789	1796	What are the new user audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MNOD_ANEW_READ	Char	8	1798	1805	What are the new auditor audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MNOD_ANEW_WRITE	Char	8	1807	1814	What are the new auditor audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MNOD_ANEW_EXEC	Char	8	1816	1823	What are the new auditor audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MNOD_REQ_S_ISGID	Yes/No	4	1825	1828	Was the S_ISGID bit requested on for this file?
MNOD_REQ_S_ISUID	Yes/No	4	1830	1833	Was the S_ISUID bit requested on for this file?
MNOD_REQ_S_ISVTX	Yes/No	4	1835	1838	Was the S_ISVTX bit requested on for this file?
MNOD_REQ_OWN_READ	Yes/No	4	1840	1843	Was the owner READ bit requested on for this file?
MNOD_REQ_OWN_WRITE	Yes/No	4	1845	1848	Was the owner WRITE bit requested on for this file?
MNOD_REQ_OWN_EXEC	Yes/No	4	1850	1853	Was the owner EXECUTE bit requested on for this file?
MNOD_REQ_GRP_READ	Yes/No	4	1855	1858	Was the group READ bit requested on for this file?
MNOD_REQ_GRP_WRITE	Yes/No	4	1860	1863	Was the group WRITE bit requested on for this file?
MNOD_REQ_GRP_EXEC	Yes/No	4	1865	1868	Was the group EXECUTE bit requested on for this file?
MNOD_REQ_OTH_READ	Yes/No	4	1870	1873	Was the other READ bit requested on for this file?

Table 93. Format of the MKNOD record extension (event code 43) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
MNOD_REQ_OTH_WRITE	Yes/No	4	1875	1878	Was the other WRITE bit requested on for this file?
MNOD_REQ_OTH_EXEC	Yes/No	4	1880	1883	Was the other EXECUTE bit requested on for this file?
MNOD_FILEPOOL	Char	8	1885	1892	SFS filepool containing the BFS file.
MNOD_FILESPACE	Char	8	1894	1901	SFS filespace containing the BFS file.
MNOD_INODE	Integer	10	1903	1912	Inode (file serial number).
MNOD_SCID	Integer	10	1914	1923	File SCID.
MNOD_DFLT_PROCESS	Yes/No	4	1925	1928	Default z/OS UNIX security environment in effect.
MNOD_UTK_NETW	Char	8	1930	1937	The port of entry network name.
MNOD_X500_SUBJECT	Char	255	1939	2193	Subject's name associated with this event.
MNOD_X500_ISSUER	Char	255	2195	2449	Issuer's name associated with this event.
MNOD_SECL	Char	8	2451	2458	Security label of the resource.
MNOD_SERV_POENAME	Char	64	2460	2523	SERVAUTH resource or profile name.
MNOD_CTX_USER	Char	510	2525	3034	Authenticated user name.
MNOD_CTX_REG	Char	255	3036	3290	Authenticated user registry name.
MNOD_CTX_HOST	Char	128	3292	3419	Authenticated user host name.
MNOD_CTX_MECH	Char	16	3421	3436	Authenticated user authentication mechanism object identifier (OID).
MNOD_IDID_USER	Char	985	3438	4422	Authenticated distributed user name.
MNOD_IDID_REG	Char	1021	4424	5444	Authenticated distributed user registry name.

Table 94. Event qualifiers for MKNOD records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Node created. There are no failure cases for this event.

The mount file system record extension

Table 95 on page 260 describes the format of a record that is created by mounting a file system.

The event qualifier that can be associated with the mounting of a file system event is shown in Table 96 on page 262.

Table 95. Format of the mount file system record extension (event code 44)					
Field name	Type	Length	Position		Comments
			Start	End	
MFS_CLASS	Char	8	282	289	Class name.
MFS_USER_NAME	Char	20	291	310	The name associated with the user ID.
MFS_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
MFS_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
MFS_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
MFS_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
MFS_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?

Table 95. Format of the mount file system record extension (event code 44) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
MFS_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
MFS_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
MFS_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
MFS_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
MFS_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
MFS_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
MFS_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
MFS_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
MFS_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
MFS_UTK_SECL	Char	8	386	393	The security label of the user.
MFS_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
MFS_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
MFS_UTK_SNODE	Char	8	413	420	The submitting node.
MFS_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
MFS_UTK_SPOE	Char	8	431	438	The port of entry.
MFS_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
MFS_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
MFS_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
MFS_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
MFS_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
MFS_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
MFS_AUDIT_CODE	Char	11	494	504	Audit function code.
MFS_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
MFS_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
MFS_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
MFS_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
MFS_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
MFS_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
MFS_PATH_NAME	Char	1023	572	1594	The requested path name.
MFS_FILE_ID	Char	32	1596	1627	File ID.
MFS_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
MFS_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
MFS_HFS_DS_NAME	Char	44	1651	1694	data set name for the mounted file system.
MFS_DCE_LINK	Char	16	1696	1711	Link to connect DCE records that originate from a single DCE request.

Table 95. Format of the mount file system record extension (event code 44) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
MFS_AUTH_TYPE	Char	13	1713	1725	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
MFS_DFLT_PROCESS	Yes/No	4	1727	1730	Default z/OS UNIX security environment in effect.
MFS_UTK_NETW	Char	8	1732	1739	The port of entry network name.
MFS_X500_SUBJECT	Char	255	1741	1995	Subject's name associated with this event.
MFS_X500_ISSUER	Char	255	1997	2251	Issuer's name associated with this event.
MFS_SERV_POENAME	Char	64	2253	2316	SERVAUTH resource or profile name.
MFS_CTX_USER	Char	510	2318	2827	Authenticated user name.
MFS_CTX_REG	Char	255	2829	3083	Authenticated user registry name.
MFS_CTX_HOST	Char	128	3085	3212	Authenticated user host name.
MFS_CTX_MECH	Char	16	3214	3229	Authenticated user authentication mechanism object identifier (OID).
MFS_IDID_USER	Char	985	3231	4215	Authenticated distributed user name.
MFS_IDID_REG	Char	1021	4217	5237	Authenticated distributed user registry name.

Table 96. Event qualifiers for mount file system records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	File system mounted. There are no failure cases for this event.

The OPENFILE record extension

Table 97 on page 262 describes the format of a record that is created by opening a file.

The event qualifier that can be associated with opening a file is shown in Table 98 on page 265.

Table 97. Format of the OPENFILE record extension (event code 45)					
Field name	Type	Length	Position		Comments
			Start	End	
OPEN_CLASS	Char	8	282	289	Class name.
OPEN_USER_NAME	Char	20	291	310	The name associated with the user ID.
OPEN_UTK_ENCR	Yes/No	4	312	315	Is the UTKEN associated with this user encrypted?
OPEN_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
OPEN_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
OPEN_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
OPEN_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
OPEN_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
OPEN_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
OPEN_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
OPEN_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
OPEN_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?

Table 97. Format of the OPENFILE record extension (event code 45) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
OPEN_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
OPEN_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
OPEN_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
OPEN_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
OPEN_UTK_SECL	Char	8	386	393	The security label of the user.
OPEN_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
OPEN_UTK_USUSER_ID	Char	8	404	411	The submitting user ID.
OPEN_UTK_SNODE	Char	8	413	420	The submitting node.
OPEN_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
OPEN_UTK_SPOE	Char	8	431	438	The port of entry.
OPEN_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
OPEN_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
OPEN_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
OPEN_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
OPEN_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
OPEN_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
OPEN_AUDIT_CODE	Char	11	494	504	Audit function code.
OPEN_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
OPEN_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
OPEN_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
OPEN_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
OPEN_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
OPEN_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
OPEN_PATH_NAME	Char	1023	572	1594	The requested path name.
OPEN_FILE_ID	Char	32	1596	1627	File ID.
OPEN_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
OPEN_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
OPEN_OLD_S_ISGID	Yes/No	4	1651	1654	Was the S_ISGID bit requested on for this file?
OPEN_OLD_S_ISUID	Yes/No	4	1656	1659	Was the S_ISUID bit requested on for this file?
OPEN_OLD_S_ISVTX	Yes/No	4	1661	1664	Was the S_ISVTX bit requested on for this file?
OPEN_OLD_OWN_READ	Yes/No	4	1666	1669	Was the owner READ bit on for this file?
OPEN_OLD_OWN_WRITE	Yes/No	4	1671	1674	Was the owner WRITE bit on for this file?
OPEN_OLD_OWN_EXEC	Yes/No	4	1676	1679	Was the owner EXECUTE bit on for this file?
OPEN_OLD_GRP_READ	Yes/No	4	1681	1684	Was the group READ bit on for this file?
OPEN_OLD_GRP_WRITE	Yes/No	4	1686	1689	Was the group WRITE bit on for this file?
OPEN_OLD_GRP_EXEC	Yes/No	4	1691	1694	Was the group EXECUTE bit on for this file?

Table 97. Format of the OPENFILE record extension (event code 45) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
OPEN_OLD_OTH_READ	Yes/No	4	1696	1699	Was the other READ bit on for this file?
OPEN_OLD_OTH_WRITE	Yes/No	4	1701	1704	Was the other WRITE bit on for this file?
OPEN_OLD_OTH_EXEC	Yes/No	4	1706	1709	Was the other EXECUTE bit on for this file?
OPEN_NEW_S_ISGID	Yes/No	4	1711	1714	Is the S_ISGID bit requested on for this file?
OPEN_NEW_S_ISUID	Yes/No	4	1716	1719	Is the S_ISUID bit requested on for this file?
OPEN_NEW_S_ISVTX	Yes/No	4	1721	1724	Is the S_ISVTX bit requested on for this file?
OPEN_NEW_OWN_READ	Yes/No	4	1726	1729	Is the owner READ bit on for this file?
OPEN_NEW_OWN_WRITE	Yes/No	4	1731	1734	Is the owner WRITE bit on for this file?
OPEN_NEW_OWN_EXEC	Yes/No	4	1736	1739	Is the owner EXECUTE bit on for this file?
OPEN_NEW_GRP_READ	Yes/No	4	1741	1744	Is the group READ bit on for this file?
OPEN_NEW_GRP_WRITE	Yes/No	4	1746	1749	Is the group WRITE bit on for this file?
OPEN_NEW_GRP_EXEC	Yes/No	4	1751	1754	Is the group EXECUTE bit on for this file?
OPEN_NEW_OTH_READ	Yes/No	4	1756	1759	Is the other READ bit on for this file?
OPEN_NEW_OTH_WRITE	Yes/No	4	1761	1764	Is the other WRITE bit on for this file?
OPEN_NEW_OTH_EXEC	Yes/No	4	1766	1769	Is the other EXECUTE bit on for this file?
OPEN_UNEW_READ	Char	8	1771	1778	What are the new user audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
OPEN_UNEW_WRITE	Char	8	1780	1787	What are the new user audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
OPEN_UNEW_EXEC	Char	8	1789	1796	What are the new user audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
OPEN_ANEW_READ	Char	8	1798	1805	What are the new auditor audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
OPEN_ANEW_WRITE	Char	8	1807	1814	What are the new auditor audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
OPEN_ANEW_EXEC	Char	8	1816	1823	What are the new auditor audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
OPEN_REQ_S_ISGID	Yes/No	4	1825	1828	Was the S_ISGID bit requested on for this file?
OPEN_REQ_S_ISUID	Yes/No	4	1830	1833	Was the S_ISUID bit requested on for this file?
OPEN_REQ_S_ISVTX	Yes/No	4	1835	1838	Was the S_ISVTX bit requested on for this file?
OPEN_REQ_OWN_READ	Yes/No	4	1840	1843	Was the owner READ bit requested on for this file?
OPEN_REQ_OWN_WRITE	Yes/No	4	1845	1848	Was the owner WRITE bit requested on for this file?
OPEN_REQ_OWN_EXEC	Yes/No	4	1850	1853	Was the owner EXECUTE bit requested on for this file?
OPEN_REQ_GRP_READ	Yes/No	4	1855	1858	Was the group READ bit requested on for this file?
OPEN_REQ_GRP_WRITE	Yes/No	4	1860	1863	Was the group WRITE bit requested on for this file?

Table 97. Format of the OPENFILE record extension (event code 45) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
OPEN_REQ_GRP_EXEC	Yes/No	4	1865	1868	Was the group EXECUTE bit requested on for this file?
OPEN_REQ_OTH_READ	Yes/No	4	1870	1873	Was the other READ bit requested on for this file?
OPEN_REQ_OTH_WRITE	Yes/No	4	1875	1878	Was the other WRITE bit requested on for this file?
OPEN_REQ_OTH_EXEC	Yes/No	4	1880	1883	Was the other EXECUTE bit requested on for this file?
OPEN_FILEPOOL	Char	8	1885	1892	SFS filepool containing the BFS file.
OPEN_FILESPACE	Char	8	1894	1901	SFS filespace containing the BFS file.
OPEN_INODE	Integer	10	1903	1912	Inode (file serial number).
OPEN_SCID	Integer	10	1914	1923	File SCID.
OPEN_DFLT_PROCESS	Yes/No	4	1925	1928	Default z/OS UNIX security environment in effect.
OPEN_UTK_NETW	Char	8	1930	1937	The port of entry network name.
OPEN_X500_SUBJECT	Char	255	1939	2193	Subject's name associated with this event.
OPEN_X500_ISSUER	Char	255	2195	2449	Issuer's name associated with this event.
OPEN_SECL	Char	8	2451	2458	Security label of the resource.
OPEN_SERV_POENAME	Char	64	2460	2523	SERVAUTH resource or profile name.
OPEN_CTX_USER	Char	510	2525	3034	Authenticated user name.
OPEN_CTX_REG	Char	255	3036	3290	Authenticated user registry name.
OPEN_CTX_HOST	Char	128	3292	3419	Authenticated user host name.
OPEN_CTX_MECH	Char	16	3421	3436	Authenticated user authentication mechanism object identifier (OID).
OPEN_IDID_USER	Char	985	3438	4422	Authenticated distributed user name.
OPEN_IDID_REG	Char	1021	4424	5444	Authenticated distributed user registry name.

Table 98. Event qualifiers for OPENFILE records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	File created. There are no failure cases for this event.

The PTRACE record extension

Table 99 on page 265 describes the format of a record that is created by the tracing of a process.

The event qualifiers that can be associated with the tracing of a process are shown in Table 100 on page 267.

Table 99. Format of the PTRACE record extension (event code 46)					
Field name	Type	Length	Position		Comments
			Start	End	
PTRC_CLASS	Char	8	282	289	Class name.
PTRC_USER_NAME	Char	20	291	310	The name associated with the user ID.
PTRC_UTK_ENCR	Yes/No	4	312	315	Is the UTKEN associated with this user encrypted?
PTRC_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?

Table 99. Format of the PTRACE record extension (event code 46) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
PTRC_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
PTRC_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
PTRC_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
PTRC_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
PTRC_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
PTRC_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
PTRC_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
PTRC_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
PTRC_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
PTRC_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
PTRC_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
PTRC_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
PTRC_UTK_SECL	Char	8	386	393	The security label of the user.
PTRC_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
PTRC_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
PTRC_UTK_SNODE	Char	8	413	420	The submitting node.
PTRC_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
PTRC_UTK_SPOE	Char	8	431	438	The port of entry.
PTRC_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
PTRC_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
PTRC_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
PTRC_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
PTRC_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
PTRC_APPC_LINK	Char	16	477	492	Key to link together APPC records.
PTRC_AUDIT_CODE	Char	11	494	504	Audit function code.
PTRC_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
PTRC_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
PTRC_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
PTRC_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
PTRC_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
PTRC_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
PTRC_TGT_REAL_UID	Integer	10	572	581	Target real z/OS UNIX user identifier (UID).
PTRC_TGT_EFF_UID	Integer	10	583	592	Target effective z/OS UNIX user identifier (UID).
PTRC_TGT_SAVED_UID	Integer	10	594	603	Target saved z/OS UNIX user identifier (UID).
PTRC_TGT_REAL_GID	Integer	10	605	614	Target real z/OS UNIX group identifier (GID).
PTRC_TGT_EFF_GID	Integer	10	616	625	Target effective z/OS UNIX group identifier (GID).
PTRC_TGT_SAVED_GID	Integer	10	627	636	Target saved z/OS UNIX group identifier (GID).

Table 99. Format of the PTRACE record extension (event code 46) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
PTRC_TGT_PID	Integer	10	638	647	Target process ID.
PTRC_DFLT_PROCESS	Yes/No	4	649	652	Default z/OS UNIX security environment in effect.
PTRC_UTK_NETW	Char	8	654	661	The port of entry network name.
PTRC_X500_SUBJECT	Char	255	663	917	Subject's name associated with this event.
PTRC_X500_ISSUER	Char	255	919	1173	Issuer's name associated with this event.
PTRC_SECL	Char	8	1175	1182	Security label of the resource.
PTRC_SERV_POENAME	Char	64	1184	1247	SERVAUTH resource or profile name.
PTRC_CTX_USER	Char	510	1249	1758	Authenticated user name.
PTRC_CTX_REG	Char	255	1760	2014	Authenticated user registry name.
PTRC_CTX_HOST	Char	128	2016	2143	Authenticated user host name.
PTRC_CTX_MECH	Char	16	2145	2160	Authenticated user authentication mechanism object identifier (OID).
PTRC_IDID_USER	Char	985	2162	3146	Authenticated distributed user name.
PTRC_IDID_REG	Char	1021	3148	4168	Authenticated distributed user registry name.

Table 100. Event qualifiers for PTRACE records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Access allowed.
NOTAUTH	01	Not authorized to trace the specified process.
INSSECL	02	Insufficient security label.

The rename file record extension

Table 89 on page 252 describes the format of a record that is created by a rename operation.

The event qualifier that can be associated with a file rename event is shown in Table 102 on page 269.

Table 101. Format of the rename file record extension (event code 47)					
Field name	Type	Length	Position		Comments
			Start	End	
RENF_CLASS	Char	8	282	289	Class name.
RENF_USER_NAME	Char	20	291	310	The name associated with the user ID.
RENF_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
RENF_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
RENF_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
RENF_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
RENF_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
RENF_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
RENF_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
RENF_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
RENF_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?

Table 101. Format of the rename file record extension (event code 47) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RENF_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
RENF_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
RENF_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
RENF_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
RENF_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
RENF_UTK_SECL	Char	8	386	393	The security label of the user.
RENF_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
RENF_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
RENF_UTK_SNODE	Char	8	413	420	The submitting node.
RENF_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
RENF_UTK_SPOE	Char	8	431	438	The port of entry.
RENF_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
RENF_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
RENF_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
RENF_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
RENF_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
RENF_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
RENF_AUDIT_CODE	Char	11	494	504	Audit function code.
RENF_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
RENF_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
RENF_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
RENF_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
RENF_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
RENF_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
RENF_PATH_NAME	Char	1023	572	1594	The requested path name.
RENF_FILE_ID	Char	32	1596	1627	File ID.
RENF_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
RENF_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
RENF_PATH2	Char	1023	1651	2673	Second requested path name.
RENF_FILE_ID2	Char	32	2675	2706	Second requested file ID.
RENF_OWNER_UID	Integer	10	2708	2717	z/OS UNIX user identifier (UID) of the owner of the deleted file.
RENF_OWNER_GID	Integer	10	2719	2728	z/OS UNIX group identifier (GID) of the owner of the deleted file.
RENF_PATH_TYPE	Char	4	2730	2733	Type of the requested path name. Valid values are "OLD" and "NEW".
RENF_LAST_DELETED	Yes/No	4	2735	2738	Was the last link deleted?

Table 101. Format of the rename file record extension (event code 47) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RENF_FILEPOOL	Char	8	2740	2747	SFS filepool containing the BFS file.
RENF_FILESPACE	Char	8	2749	2756	SFS filespace containing the BFS file.
RENF_INODE	Integer	10	2758	2767	Inode (file serial number).
RENF_SCID	Integer	10	2769	2778	File SCID.
RENF_FILEPOOL2	Char	8	2780	2787	SFS filepool containing the second BFS file.
RENF_FILESPACE2	Char	8	2789	2796	SFS filespace containing the second BFS file.
RENF_INODE2	Integer	10	2798	2807	Second Inode (file serial number).
RENF_SCID2	Integer	10	2809	2818	Second file SCID.
RENF_DCE_LINK	Char	16	2820	2835	Link to connect DCE records that originate from a single DCE request.
RENF_AUTH_TYPE	Char	13	2837	2849	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
RENF_DFLT_PROCESS	Yes/No	4	2851	2854	Default z/OS UNIX security environment in effect.
RENF_UTK_NETW	Char	8	2856	2863	The port of entry network name.
RENF_X500_SUBJECT	Char	255	2865	3119	Subject's name associated with this event.
RENF_X500_ISSUER	Char	255	3121	3375	Issuer's name associated with this event.
RENF_SERV_POENAME	Char	64	3377	3440	SERVAUTH resource or profile name.
RENF_CTX_USER	Char	510	3442	3951	Authenticated user name.
RENF_CTX_REG	Char	255	3953	4207	Authenticated user registry name.
RENF_CTX_HOST	Char	128	4209	4336	Authenticated user host name.
RENF_CTX_MECH	Char	16	4338	4353	Authenticated user authentication mechanism object identifier (OID).
RENF_IDID_USER	Char	985	4355	5339	Authenticated distributed user name.
RENF_IDID_REG	Char	1021	5341	6361	Authenticated distributed user registry name.

Table 102. Event qualifiers for rename file records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	File renamed. There are no failure cases for this event.

The RMDIR record extension

Table 103 on page 269 describes the format of a record that is created by removing a directory.

The event qualifier that can be associated with removing a directory is shown in Table 104 on page 271.

Table 103. Format of the RMDIR record extension (event code 48)					
Field name	Type	Length	Position		Comments
			Start	End	
RDIR_CLASS	Char	8	282	289	Class name.
RDIR_USER_NAME	Char	20	291	310	The name associated with the user ID.
RDIR_UTK_ENCR	Yes/No	4	312	315	Is the UTKEN associated with this user encrypted?
RDIR_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?

Table 103. Format of the RMDIR record extension (event code 48) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RDIR_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
RDIR_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
RDIR_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
RDIR_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
RDIR_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
RDIR_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
RDIR_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
RDIR_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
RDIR_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
RDIR_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
RDIR_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
RDIR_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
RDIR_UTK_SECL	Char	8	386	393	The security label of the user.
RDIR_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
RDIR_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
RDIR_UTK_SNODE	Char	8	413	420	The submitting node.
RDIR_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
RDIR_UTK_SPOE	Char	8	431	438	The port of entry.
RDIR_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
RDIR_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
RDIR_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
RDIR_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
RDIR_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
RDIR_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
RDIR_AUDIT_CODE	Char	11	494	504	Audit function code.
RDIR_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
RDIR_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
RDIR_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
RDIR_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
RDIR_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
RDIR_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
RDIR_PATH_NAME	Char	1023	572	1594	The requested path name.
RDIR_FILE_ID	Char	32	1596	1627	File ID.
RDIR_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
RDIR_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
RDIR_FILEPOOL	Char	8	1651	1658	SFS filepool containing the BFS file.

Table 103. Format of the RMDIR record extension (event code 48) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RDIR_FILESPACE	Char	8	1660	1667	SFS filespace containing the BFS file.
RDIR_INODE	Integer	10	1669	1678	Inode (file serial number).
RDIR_SCID	Integer	10	1680	1689	File SCID.
RDIR_DCE_LINK	Char	16	1691	1706	Link to connect DCE records that originate from a single DCE request.
RDIR_AUTH_TYPE	Char	13	1708	1720	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
RDIR_DFLT_PROCESS	Yes/No	4	1722	1725	Default z/OS UNIX security environment in effect.
RDIR_UTK_NETW	Char	8	1727	1734	The port of entry network name.
RDIR_X500_SUBJECT	Char	255	1736	1990	Subject's name associated with this event.
RDIR_X500_ISSUER	Char	255	1992	2246	Issuer's name associated with this event.
RDIR_SERV_POENAME	Char	64	2248	2311	SERVAUTH resource or profile name.
RDIR_CTX_USER	Char	510	2313	2822	Authenticated user name.
RDIR_CTX_REG	Char	255	2824	3078	Authenticated user registry name.
RDIR_CTX_HOST	Char	128	3080	3207	Authenticated user host name.
RDIR_CTX_MECH	Char	16	3209	3224	Authenticated user authentication mechanism object identifier (OID).
RDIR_IDID_USER	Char	985	3226	4210	Authenticated distributed user name.
RDIR_IDID_REG	Char	1021	4212	5232	Authenticated distributed user registry name.

Table 104. Event qualifiers for RMDIR records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Directory removed. There are no failure cases for this event.

The SETEGID record extension

Table 105 on page 271 describes the format of a record that is created by the setting of an effective z/OS UNIX group identifier (GID).

The event qualifiers that can be associated with setting the effective z/OS UNIX group identifier (GID) are shown in [Table 106 on page 273](#).

Table 105. Format of the SETEGID record extension (event code 49)					
Field name	Type	Length	Position		Comments
			Start	End	
SEGI_CLASS	Char	8	282	289	Class name.
SEGI_USER_NAME	Char	20	291	310	The name associated with the user ID.
SEGI_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
SEGI_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
SEGI_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
SEGI_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
SEGI_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?

Table 105. Format of the SETEGID record extension (event code 49) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
SEGI_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
SEGI_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
SEGI_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
SEGI_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
SEGI_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
SEGI_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
SEGI_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
SEGI_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
SEGI_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
SEGI_UTK_SECL	Char	8	386	393	The security label of the user.
SEGI_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
SEGI_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
SEGI_UTK_SNODE	Char	8	413	420	The submitting node.
SEGI_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
SEGI_UTK_SPOE	Char	8	431	438	The port of entry.
SEGI_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
SEGI_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
SEGI_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
SEGI_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
SEGI_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
SEGI_APPC_LINK	Char	16	477	492	Key to link together APPC records.
SEGI_AUDIT_CODE	Char	11	494	504	Audit function code.
SEGI_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
SEGI_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
SEGI_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
SEGI_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
SEGI_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
SEGI_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
SEGI_NEW_REAL_GID	Integer	10	572	581	New real z/OS UNIX group identifier (GID).
SEGI_NEW_EFF_GID	Integer	10	583	592	New effective z/OS UNIX group identifier (GID).
SEGI_NEW_SAVED_GID	Integer	10	594	603	New saved z/OS UNIX group identifier (GID).
SEGI_GID	Integer	10	605	614	The z/OS UNIX group identifier (GID) input parameter.
SEGI_DFLT_PROCESS	Yes/No	4	616	619	Default z/OS UNIX security environment in effect.
SEGI_UTK_NETW	Char	8	621	628	The port of entry network name.
SEGI_X500_SUBJECT	Char	255	630	884	Subject's name associated with this event.
SEGI_X500_ISSUER	Char	255	886	1140	Issuer's name associated with this event.
SEGI_SERV_POENAME	Char	64	1142	1205	SERVAUTH resource or profile name.

Table 105. Format of the SETEGID record extension (event code 49) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
SEGI_CTX_USER	Char	510	1207	1716	Authenticated user name.
SEGI_CTX_REG	Char	255	1718	1972	Authenticated user registry name.
SEGI_CTX_HOST	Char	128	1974	2101	Authenticated user host name.
SEGI_CTX_MECH	Char	16	2103	2118	Authenticated user authentication mechanism object identifier (OID).
SEGI_IDID_USER	Char	985	2120	3104	Authenticated distributed user name.
SEGI_IDID_REG	Char	1021	3106	4126	Authenticated distributed user registry name.

Table 106. Event qualifiers for SETEGID records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful change of effective z/OS UNIX group identifier (GID).
NOTAUTH	01	Not authorized to set the effective z/OS UNIX group identifier (GID).

The SETEUID record extension

Table 107 on page 273 describes the format of a record that is created by the setting of an effective z/OS UNIX user identifier (UID).

The event qualifiers that can be associated with setting the effective z/OS UNIX user identifier (UID) are shown in [Table 108 on page 274](#).

Table 107. Format of the SETEUID record extension (event code 50)					
Field name	Type	Length	Position		Comments
			Start	End	
SEUI_CLASS	Char	8	282	289	Class name.
SEUI_USER_NAME	Char	20	291	310	The name associated with the user ID.
SEUI_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
SEUI_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
SEUI_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
SEUI_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
SEUI_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
SEUI_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
SEUI_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
SEUI_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
SEUI_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
SEUI_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
SEUI_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
SEUI_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
SEUI_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
SEUI_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
SEUI_UTK_SECL	Char	8	386	393	The security label of the user.

Table 107. Format of the SETEUID record extension (event code 50) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
SEUI_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
SEUI_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
SEUI_UTK_SNODE	Char	8	413	420	The submitting node.
SEUI_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
SEUI_UTK_SPOE	Char	8	431	438	The port of entry.
SEUI_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
SEUI_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
SEUI_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
SEUI_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
SEUI_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
SEUI_APPC_LINK	Char	16	477	492	Key to link together APPC records.
SEUI_AUDIT_CODE	Char	11	494	504	Audit function code.
SEUI_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
SEUI_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
SEUI_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
SEUI_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
SEUI_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
SEUI_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
SEUI_NEW_REAL_UID	Integer	10	572	581	New real z/OS UNIX user identifier (UID).
SEUI_NEW_EFF_UID	Integer	10	583	592	New effective z/OS UNIX user identifier (UID).
SEUI_NEW_SAVED_UID	Integer	10	594	603	New saved z/OS UNIX user identifier (UID).
SEUI_UID	Integer	10	605	614	The z/OS UNIX user identifier (UID) input parameter.
SEUI_DFLT_PROCESS	Yes/No	4	616	619	Default z/OS UNIX security environment in effect.
SEUI_UTK_NETW	Char	8	621	628	The port of entry network name.
SEUI_X500_SUBJECT	Char	255	630	884	Subject's name associated with this event.
SEUI_X500_ISSUER	Char	255	886	1140	Issuer's name associated with this event.
SEUI_SERV_POENAME	Char	64	1142	1205	SERVAUTH resource or profile name.
SEUI_CTX_USER	Char	510	1207	1716	Authenticated user name.
SEUI_CTX_REG	Char	255	1718	1972	Authenticated user registry name.
SEUI_CTX_HOST	Char	128	1974	2101	Authenticated user host name.
SEUI_CTX_MECH	Char	16	2103	2118	Authenticated user authentication mechanism object identifier (OID).
SEUI_IDID_USER	Char	985	2120	3104	Authenticated distributed user name.
SEUI_IDID_REG	Char	1021	3106	4126	Authenticated distributed user registry name.

Table 108. Event qualifiers for SETEUID records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful change of z/OS UNIX user identifiers (UIDs).

Table 108. Event qualifiers for SETEUID records (continued)

Event qualifier	Event qualifier number	Event description
NOTAUTH	01	Not authorized to set the effective z/OS UNIX user identifier (UID).

The SETGID record extension

Table 109 on page 275 describes the format of a record that is created by the setting of a z/OS UNIX group identifier (GID).

The event qualifiers that can be associated with setting the z/OS UNIX group identifier (GID) are shown in Table 110 on page 276.

Table 109. Format of the SETGID record extension (event code 51)

Field name	Type	Length	Position		Comments
			Start	End	
SGI_CLASS	Char	8	282	289	Class name.
SGI_USER_NAME	Char	20	291	310	The name associated with the user ID.
SGI_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
SGI_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
SGI_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
SGI_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
SGI_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
SGI_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
SGI_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
SGI_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
SGI_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
SGI_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
SGI_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
SGI_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
SGI_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
SGI_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
SGI_UTK_SECL	Char	8	386	393	The security label of the user.
SGI_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
SGI_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
SGI_UTK_SNODE	Char	8	413	420	The submitting node.
SGI_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
SGI_UTK_SPOE	Char	8	431	438	The port of entry.
SGI_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
SGI_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
SGI_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
SGI_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
SGI_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?

Table 109. Format of the SETGID record extension (event code 51) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
SGI_APPC_LINK	Char	16	477	492	Key to link together APPC records.
SGI_AUDIT_CODE	Char	11	494	504	Audit function code.
SGI_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
SGI_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
SGI_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
SGI_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
SGI_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
SGI_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
SGI_NEW_REAL_GID	Integer	10	572	581	New real z/OS UNIX group identifier (GID).
SGI_NEW_EFF_GID	Integer	10	583	592	New effective z/OS UNIX group identifier (GID).
SGI_NEW_SAVED_GID	Integer	10	594	603	New saved z/OS UNIX group identifier (GID).
SGI_GID	Integer	10	605	614	The z/OS UNIX group identifier (GID) input parameter.
SGI_DFLT_PROCESS	Yes/No	4	616	619	Default z/OS UNIX security environment in effect.
SGI_UTK_NETW	Char	8	621	628	The port of entry network name.
SGI_X500_SUBJECT	Char	255	630	884	Subject's name associated with this event.
SGI_X500_ISSUER	Char	255	886	1140	Issuer's name associated with this event.
SGI_SERV_POENAME	Char	64	1142	1205	SERVAUTH resource or profile name.
SGI_CTX_USER	Char	510	1207	1716	Authenticated user name.
SGI_CTX_REG	Char	255	1718	1972	Authenticated user registry name.
SGI_CTX_HOST	Char	128	1974	2101	Authenticated user host name.
SGI_CTX_MECH	Char	16	2103	2118	Authenticated user authentication mechanism object identifier (OID).
SGI_IDID_USER	Char	985	2120	3104	Authenticated distributed user name.
SGI_IDID_REG	Char	1021	3106	4126	Authenticated distributed user registry name.

Table 110. Event qualifiers for SETGID records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful change of z/OS UNIX group identifier (GID).
NOTAUTH	01	Not authorized to set the z/OS UNIX group identifier (GID).

The SETUID record extension

Table 111 on page 277 describes the format of a record that is created by the setting of a z/OS UNIX user identifier (UID).

The event qualifiers that can be associated with setting the effective z/OS UNIX user identifier (UID) are shown in Table 112 on page 278.

Table 111. Format of the SETUID record extension (event code 52)					
Field name	Type	Length	Position		Comments
			Start	End	
SUI_CLASS	Char	8	282	289	Class name.
SUI_USER_NAME	Char	20	291	310	The name associated with the user ID.
SUI_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
SUI_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
SUI_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
SUI_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
SUI_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
SUI_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
SUI_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
SUI_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
SUI_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
SUI_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
SUI_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
SUI_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
SUI_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
SUI_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
SUI_UTK_SECL	Char	8	386	393	The security label of the user.
SUI_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
SUI_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
SUI_UTK_SNODE	Char	8	413	420	The submitting node.
SUI_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
SUI_UTK_SPOE	Char	8	431	438	The port of entry.
SUI_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
SUI_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
SUI_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
SUI_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
SUI_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
SUI_APPC_LINK	Char	16	477	492	Key to link together APPC records.
SUI_AUDIT_CODE	Char	11	494	504	Audit function code.
SUI_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
SUI_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
SUI_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
SUI_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
SUI_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
SUI_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
SUI_NEW_REAL_UID	Integer	10	572	581	New real z/OS UNIX user identifier (UID).
SUI_NEW_EFF_UID	Integer	10	583	592	New effective z/OS UNIX user identifier (UID).

Table 111. Format of the SETUID record extension (event code 52) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
SUI_NEW_SAVED_UID	Integer	10	594	603	New saved z/OS UNIX user identifier (UID).
SUI_UID	Integer	10	605	614	The z/OS UNIX user identifier (UID) input parameter.
SUI_DFLT_PROCESS	Yes/No	4	616	619	Default z/OS UNIX security environment in effect.
SUI_UTK_NETW	Char	8	621	628	The port of entry network name.
SUI_X500_SUBJECT	Char	255	630	884	Subject's name associated with this event.
SUI_X500_ISSUER	Char	255	886	1140	Issuer's name associated with this event.
SUI_SERV_POENAME	Char	64	1142	1205	SERVAUTH resource or profile name.
SUI_CTX_USER	Char	510	1207	1716	Authenticated user name.
SUI_CTX_REG	Char	255	1718	1972	Authenticated user registry name.
SUI_CTX_HOST	Char	128	1974	2101	Authenticated user host name.
SUI_CTX_MECH	Char	16	2103	2118	Authenticated user authentication mechanism object identifier (OID).
SUI_IDID_USER	Char	985	2120	3104	Authenticated distributed user name.
SUI_IDID_REG	Char	1021	3106	4126	Authenticated distributed user registry name.

Table 112. Event qualifiers for SETUID records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful change of z/OS UNIX user identifier (UID).
NOTAUTH	01	Not authorized to set the z/OS UNIX user identifier (UID).

The SYMLINK record extension

Table 113 on page 278 describes the format of a record that is created by a SYMLINK operation.

The event qualifier that can be associated with a SYMLINK event is shown in Table 114 on page 280.

Table 113. Format of the SYMLINK record extension (event code 53)					
Field name	Type	Length	Position		Comments
			Start	End	
SYML_CLASS	Char	8	282	289	Class name.
SYML_USER_NAME	Char	20	291	310	The name associated with the user ID.
SYML_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
SYML_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
SYML_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
SYML_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
SYML_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
SYML_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
SYML_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
SYML_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
SYML_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?

Table 113. Format of the SYMLINK record extension (event code 53) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
SYML_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
SYML_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
SYML_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
SYML_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
SYML_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
SYML_UTK_SECL	Char	8	386	393	The security label of the user.
SYML_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
SYML_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
SYML_UTK_SNODE	Char	8	413	420	The submitting node.
SYML_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
SYML_UTK_SPOE	Char	8	431	438	The port of entry.
SYML_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
SYML_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
SYML_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
SYML_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
SYML_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
SYML_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
SYML_AUDIT_CODE	Char	11	494	504	Audit function code.
SYML_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
SYML_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
SYML_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
SYML_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
SYML_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
SYML_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
SYML_PATH_NAME	Char	1023	572	1594	The requested path name.
SYML_FILE_ID	Char	32	1596	1627	File ID.
SYML_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
SYML_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
SYML_SYMLINK_DATA	Char	1023	1651	2673	Content of SYMLINK.
SYML_FILEPOOL	Char	8	2675	2682	SFS filepool containing the BFS file.
SYML_FILESPACE	Char	8	2684	2691	SFS filespace containing the BFS file.
SYML_INODE	Integer	10	2693	2702	Inode (file serial number).
SYML_SCID	Integer	10	2704	2713	File SCID.
SYML_DFLT_PROCESS	Char	1	2715	2715	Default z/OS UNIX security environment in effect.
SYML_UTK_NETW	Char	8	2720	2727	The port of entry network name.
SYML_X500_SUBJECT	Char	255	2729	2983	Subject's name associated with this event.

Table 113. Format of the SYMLINK record extension (event code 53) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
SYML_X500_ISSUER	Char	255	2985	3239	Issuer's name associated with this event.
SYML_SECL	Char	8	3241	3248	Security label of the resource.
SYML_SERV_POENAME	Char	64	3250	3313	SERVAUTH resource or profile name.
SYML_CTX_USER	Char	510	3315	3824	Authenticated user name.
SYML_CTX_REG	Char	255	3826	4080	Authenticated user registry name.
SYML_CTX_HOST	Char	128	4082	4209	Authenticated user host name.
SYML_CTX_MECH	Char	16	4211	4226	Authenticated user authentication mechanism object identifier (OID).
SYML_IDID_USER	Char	985	4228	5212	Authenticated distributed user name.
SYML_IDID_REG	Char	1021	5214	6234	Authenticated distributed user registry name.

Table 114. Event qualifiers for SYMLINK records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful SYMLINK. There are no failure cases for this event.

The UNLINK record extension

Table 115 on page 280 describes the format of a record that is created by an UNLINK operation.

The event qualifier that can be associated with an UNLINK event is shown in Table 116 on page 282.

Table 115. Format of the UNLINK record extension (event code 54)					
Field name	Type	Length	Position		Comments
			Start	End	
UNL_CLASS	Char	8	282	289	Class name.
UNL_USER_NAME	Char	20	291	310	The name associated with the user ID.
UNL_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
UNL_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
UNL_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
UNL_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
UNL_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
UNL_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
UNL_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
UNL_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
UNL_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
UNL_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
UNL_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
UNL_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
UNL_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
UNL_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?

Table 115. Format of the UNLINK record extension (event code 54) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
UNL_UTK_SECL	Char	8	386	393	The security label of the user.
UNL_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
UNL_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
UNL_UTK_SNODE	Char	8	413	420	The submitting node.
UNL_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
UNL_UTK_SPOE	Char	8	431	438	The port of entry.
UNL_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
UNL_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
UNL_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
UNL_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
UNL_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
UNL_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
UNL_AUDIT_CODE	Char	11	494	504	Audit function code.
UNL_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
UNL_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
UNL_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
UNL_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
UNL_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
UNL_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
UNL_PATH_NAME	Char	1023	572	1594	The requested path name.
UNL_FILE_ID	Char	32	1596	1627	File ID.
UNL_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
UNL_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
UNL_LAST_DELETED	Yes/No	4	1651	1654	Was the last link deleted?
UNL_FILEPOOL	Char	8	1656	1663	SFS filepool containing the BFS file.
UNL_FILESPACE	Char	8	1665	1672	SFS filespace containing the BFS file.
UNL_INODE	Integer	10	1674	1683	Inode (file serial number).
UNL_SCID	Integer	10	1685	1694	File SCID.
UNL_DCE_LINK	Char	16	1696	1711	Link to connect DCE records that originate from a single DCE request.
UNL_AUTH_TYPE	Char	13	1713	1725	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
UNL_DFLT_PROCESS	Yes/No	4	1727	1730	Default z/OS UNIX security environment in effect.
UNL_UTK_NETW	Char	8	1732	1739	The port of entry network name.
UNL_X500_SUBJECT	Char	255	1741	1995	Subject's name associated with this event.
UNL_X500_ISSUER	Char	255	1997	2251	Issuer's name associated with this event.
UNL_SERV_POENAME	Char	64	2253	2316	SERVAUTH resource or profile name.

Table 115. Format of the UNLINK record extension (event code 54) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
UNL_CTX_USER	Char	510	2318	2827	Authenticated user name.
UNL_CTX_REG	Char	255	2829	3083	Authenticated user registry name.
UNL_CTX_HOST	Char	128	3085	3212	Authenticated user host name.
UNL_CTX_MECH	Char	16	3214	3229	Authenticated user authentication mechanism object identifier (OID).
UNL_IDID_USER	Char	985	3231	4215	Authenticated distributed user name.
UNL_IDID_REG	Char	1021	4217	5237	Authenticated distributed user registry name.

Table 116. Event qualifiers for UNLINK records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful UNLINK. Failures are logged as check access event types.

The unmount file system record extension

Table 117 on page 282 describes the format of a record that is created unmounting a file system.

The event qualifier that can be associated with the unmounting of a file system is shown in Table 118 on page 284.

Table 117. Format of the unmount file system record extension (event code 55)					
Field name	Type	Length	Position		Comments
			Start	End	
UFS_CLASS	Char	8	282	289	Class name.
UFS_USER_NAME	Char	20	291	310	The name associated with the user ID.
UFS_UTK_ENCR	Yes/No	4	312	315	Is the UTKEN associated with this user encrypted?
UFS_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
UFS_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
UFS_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
UFS_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
UFS_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
UFS_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
UFS_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
UFS_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
UFS_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
UFS_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
UFS_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
UFS_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
UFS_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
UFS_UTK_SECL	Char	8	386	393	The security label of the user.
UFS_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
UFS_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.

Table 117. Format of the unmount file system record extension (event code 55) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
UFS_UTK_SNODE	Char	8	413	420	The submitting node.
UFS_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
UFS_UTK_SPOE	Char	8	431	438	The port of entry.
UFS_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
UFS_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
UFS_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
UFS_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
UFS_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
UFS_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
UFS_AUDIT_CODE	Char	11	494	504	Audit function code.
UFS_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
UFS_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
UFS_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
UFS_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
UFS_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
UFS_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
UFS_PATH_NAME	Char	1023	572	1594	The requested path name.
UFS_FILE_ID	Char	32	1596	1627	File ID.
UFS_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
UFS_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
UFS_HFS_DS_NAME	Char	44	1651	1694	Data set name for the mounted file system.
UFS_DCE_LINK	Char	16	1696	1711	Link to connect DCE records that originate from a single DCE request.
UFS_AUTH_TYPE	Char	13	1713	1725	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
UFS_DFLT_PROCESS	Yes/No	4	1727	1730	Default z/OS UNIX security environment in effect.
UFS_UTK_NETW	Char	8	1732	1739	The port of entry network name.
UFS_X500_SUBJECT	Char	255	1741	1995	Subject's name associated with this event.
UFS_X500_ISSUER	Char	255	1997	2251	Issuer's name associated with this event.
UFS_SERV_POENAME	Char	64	2253	2316	SERVAUTH resource or profile name.
UFS_CTX_USER	Char	510	2318	2827	Authenticated user name.
UFS_CTX_REG	Char	255	2829	3083	Authenticated user registry name.
UFS_CTX_HOST	Char	128	3085	3212	Authenticated user host name.
UFS_CTX_MECH	Char	16	3214	3229	Authenticated user authentication mechanism object identifier (OID).
UFS_IDID_USER	Char	985	3231	4215	Authenticated distributed user name.
UFS_IDID_REG	Char	1021	4217	5237	Authenticated distributed user registry name.

Table 118. Event qualifiers for unmount file system records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	Unmount successful. Failures are logged as CKPRIV events.

The check file owner record extension

Table 119 on page 284 describes the format of a record that is created by checking the owner of a file.

The event qualifiers that can be associated with checking a file's owner are shown in Table 120 on page 285.

Table 119. Format of the check file owner record extension (event code 56)

Field name	Type	Length	Position		Comments
			Start	End	
CFOW_CLASS	Char	8	282	289	Class name.
CFOW_USER_NAME	Char	20	291	310	The name associated with the user ID.
CFOW_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
CFOW_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
CFOW_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
CFOW_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
CFOW_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
CFOW_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
CFOW_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
CFOW_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
CFOW_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
CFOW_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
CFOW_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
CFOW_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
CFOW_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
CFOW_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
CFOW_UTK_SECL	Char	8	386	393	The security label of the user.
CFOW_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
CFOW_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
CFOW_UTK_SNODE	Char	8	413	420	The submitting node.
CFOW_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
CFOW_UTK_SPOE	Char	8	431	438	The port of entry.
CFOW_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
CFOW_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
CFOW_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
CFOW_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
CFOW_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
CFOW_APPC_LINK	Char	16	477	492	Key to link together APPC records.

Table 119. Format of the check file owner record extension (event code 56) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
CFOW_AUDIT_CODE	Char	11	494	504	Audit function code.
CFOW_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
CFOW_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
CFOW_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
CFOW_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
CFOW_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
CFOW_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
CFOW_PATH_NAME	Char	1023	572	1594	The requested path name.
CFOW_FILE_ID	Char	32	1596	1627	File ID.
CFOW_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
CFOW_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
CFOW_FILEPOOL	Char	8	1651	1658	SFS filepool containing the BFS file.
CFOW_FILESPACE	Char	8	1660	1667	SFS filespace containing the BFS file.
CFOW_INODE	Integer	10	1669	1678	Inode (file serial number).
CFOW_SCID	Integer	10	1680	1689	File SCID.
CFOW_DCE_LINK	Char	16	1691	1706	Link to connect DCE records that originate from a single DCE request.
CFOW_AUTH_TYPE	Char	13	1708	1720	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
CFOW_DFLT_PROCESS	Yes/No	4	1722	1725	Default z/OS UNIX security environment in effect.
CFOW_UTK_NETW	Char	8	1727	1734	The port of entry network name.
CFOW_X500_SUBJECT	Char	255	1736	1990	Subject's name associated with this event.
CFOW_X500_ISSUER	Char	255	1992	2246	Issuer's name associated with this event.
CFOW_SECL	Char	8	2248	2255	Security label of the resource.
CFOW_SERV_POENAME	Char	64	2257	2320	SERVAUTH resource or profile name.
CFOW_CTX_USER	Char	510	2322	2831	Authenticated user name.
CFOW_CTX_REG	Char	255	2833	3087	Authenticated user registry name.
CFOW_CTX_HOST	Char	128	3089	3216	Authenticated user host name.
CFOW_CTX_MECH	Char	16	3218	3233	Authenticated user authentication mechanism object identifier (OID).
CFOW_IDID_USER	Char	985	3235	4219	Authenticated distributed user name.
CFOW_IDID_REG	Char	1021	4221	5241	Authenticated distributed user registry name.

Table 120. Event qualifiers for check file owner records		
Event qualifier	Event qualifier number	Event description
OWNER	00	The user is the owner.
NOTOWNER	01	The user is not the owner.
INSSECL	02	Insufficient security label.

The check privilege record extension

Table 121 on page 286 describes the format of a record that is created by checking a user's privileges.

The event qualifiers that can be associated with checking a user's privileges are shown in Table 122 on page 287.

Table 121. Format of the check privileges record extension (event code 57)					
Field name	Type	Length	Position		Comments
			Start	End	
CPRV_CLASS	Char	8	282	289	Class name.
CPRV_USER_NAME	Char	20	291	310	The name associated with the user ID.
CPRV_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
CPRV_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
CPRV_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
CPRV_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
CPRV_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
CPRV_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
CPRV_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
CPRV_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
CPRV_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
CPRV_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
CPRV_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
CPRV_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
CPRV_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
CPRV_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
CPRV_UTK_SECL	Char	8	386	393	The security label of the user.
CPRV_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
CPRV_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
CPRV_UTK_SNODE	Char	8	413	420	The submitting node.
CPRV_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
CPRV_UTK_SPOE	Char	8	431	438	The port of entry.
CPRV_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
CPRV_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
CPRV_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
CPRV_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
CPRV_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
CPRV_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
CPRV_AUDIT_CODE	Char	11	494	504	Audit function code.
CPRV_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
CPRV_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
CPRV_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).

Table 121. Format of the check privileges record extension (event code 57) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
CPRV_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
CPRV_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
CPRV_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
CPRV_DCE_LINK	Char	16	572	587	Link to connect DCE records that originate from a single DCE request.
CPRV_AUTH_TYPE	Char	13	589	601	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
CPRV_DFLT_PROCESS	Yes/No	4	603	606	Default z/OS UNIX security environment in effect.
CPRV_UTK_NETW	Char	8	608	615	The port of entry network name.
CPRV_X500_SUBJECT	Char	255	617	871	Subject's name associated with this event.
CPRV_X500_ISSUER	Char	255	873	1127	Issuer's name associated with this event.
CPRV_SERV_POENAME	Char	64	1129	1192	SERVAUTH resource or profile name.
CPRV_CTX_USER	Char	510	1194	1703	Authenticated user name.
CPRV_CTX_REG	Char	255	1705	1959	Authenticated user registry name.
CPRV_CTX_HOST	Char	128	1961	2088	Authenticated user host name.
CPRV_CTX_MECH	Char	16	2090	2105	Authenticated user authentication mechanism object identifier (OID).
CPRV_IDID_USER	Char	985	2107	3091	Authenticated distributed user name.
CPRV_IDID_REG	Char	1021	3093	4113	Authenticated distributed user registry name.

Table 122. Event qualifiers for check privileges records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	User is authorized.
NOTAUTH	01	The user is not authorized to the function.

The open subsidiary TTY record extension

Table 123 on page 287 describes the format of a record that is created by the opening of a subsidiary TTY.

The event qualifiers that can be associated with open subsidiary TTY records are shown in [Table 124 on page 289](#).

Table 123. Format of the open subsidiary TTY record extension (event code 58)					
Field name	Type	Length	Position		Comments
			Start	End	
OSTY_CLASS	Char	8	282	289	Class name.
OSTY_USER_NAME	Char	20	291	310	The name associated with the user ID.
OSTY_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
OSTY_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
OSTY_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
OSTY_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
OSTY_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?

Table 123. Format of the open subsidiary TTY record extension (event code 58) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
OSTY_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
OSTY_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
OSTY_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
OSTY_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
OSTY_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
OSTY_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
OSTY_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
OSTY_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
OSTY_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
OSTY_UTK_SECL	Char	8	386	393	The security label of the user.
OSTY_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
OSTY_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
OSTY_UTK_SNODE	Char	8	413	420	The submitting node.
OSTY_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
OSTY_UTK_SPOE	Char	8	431	438	The port of entry.
OSTY_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
OSTY_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
OSTY_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
OSTY_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
OSTY_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
OSTY_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
OSTY_AUDIT_CODE	Char	11	494	504	Audit function code.
OSTY_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
OSTY_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
OSTY_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
OSTY_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
OSTY_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
OSTY_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
OSTY_TGT_REAL_UID	Integer	10	572	581	Target real z/OS UNIX user identifier (UID).
OSTY_TGT_EFF_UID	Integer	10	583	592	Target effective z/OS UNIX user identifier (UID).
OSTY_TGT_SAV_UID	Integer	10	594	603	Target saved z/OS UNIX user identifier (UID).
OSTY_TGT_PID	Integer	10	605	614	Target process ID.
OSTY_DFLT_PROCESS	Yes/No	4	616	619	Default z/OS UNIX security environment in effect.
OSTY_UTK_NETW	Char	8	621	628	The port of entry network name.
OSTY_X500_SUBJECT	Char	255	630	884	Subject's name associated with this event.
OSTY_X500_ISSUER	Char	255	886	1140	Issuer's name associated with this event.
OSTY_SERV_POENAME	Char	64	1142	1205	SERVAUTH resource or profile name.

Table 123. Format of the open subsidiary TTY record extension (event code 58) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
OSTY_CTX_USER	Char	510	1207	1716	Authenticated user name.
OSTY_CTX_REG	Char	255	1718	1972	Authenticated user registry name.
OSTY_CTX_HOST	Char	128	1974	2101	Authenticated user host name.
OSTY_CTX_MECH	Char	16	2103	2118	Authenticated user authentication mechanism object identifier (OID).
OSTY_IDID_USER	Char	985	2120	3104	Authenticated distributed user name.
OSTY_IDID_REG	Char	1021	3106	4126	Authenticated distributed user registry name.

Table 124. Event qualifiers for open subsidiary TTY records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Access allowed.
NOTAUTH	01	Not authorized to the specified process.

The RACLINK command record extension

Table 125 on page 289 describes the format of a record that is created by a RACLINK command.

The event qualifiers that can be associated with a RACLINK command are shown in Table 126 on page 292.

Table 125. Format of the RACLINK command record extension (event code 59)					
Field name	Type	Length	Position		Comments
			Start	End	
RACL_USER_NAME	Char	20	282	301	The name associated with the user ID.
RACL_UTK_ENCR	Yes/No	4	303	306	Is the UTOKEN associated with this user encrypted?
RACL_UTK_PRE19	Yes/No	4	308	311	Is this a pre-1.9 token?
RACL_UTK_VERPROF	Yes/No	4	313	316	Is the VERIFYX propagation flag set?
RACL_UTK_NJEUNUSR	Yes/No	4	318	321	Is this the NJE undefined user?
RACL_UTK_LOGUSR	Yes/No	4	323	326	Is UAUDIT specified for this user?
RACL_UTK_SPECIAL	Yes/No	4	328	331	Is this a SPECIAL user?
RACL_UTK_DEFAULT	Yes/No	4	333	336	Is this a default token?
RACL_UTK_UNKNUSR	Yes/No	4	338	341	Is this an undefined user?
RACL_UTK_ERROR	Yes/No	4	343	346	Is this user token in error?
RACL_UTK_TRUSTED	Yes/No	4	348	351	Is this user a part of the trusted computing base (TCB)?
RACL_UTK_SESSTYPE	Char	8	353	360	The session type of this session.
RACL_UTK_SURROGAT	Yes/No	4	362	365	Is this a surrogate user?
RACL_UTK_REMOTE	Yes/No	4	367	370	Is this a remote job?
RACL_UTK_PRIV	Yes/No	4	372	375	Is this a privileged user ID?
RACL_UTK_SECL	Char	8	377	384	The security label of the user.
RACL_UTK_EXECNODE	Char	8	386	393	The execution node of the work.
RACL_UTK_SUSER_ID	Char	8	395	402	The submitting user ID.

<i>Table 125. Format of the RACLINK command record extension (event code 59) (continued)</i>					
Field name	Type	Length	Position		Comments
			Start	End	
RACL_UTK_SNODE	Char	8	404	411	The submitting node.
RACL_UTK_SGRP_ID	Char	8	413	420	The submitting group name.
RACL_UTK_SPOE	Char	8	422	429	The port of entry.
RACL_UTK_SPCCLASS	Char	8	431	438	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
RACL_UTK_USER_ID	Char	8	440	447	User ID associated with the record.
RACL_UTK_GRP_ID	Char	8	449	456	Group name associated with the record.
RACL_UTK_DFT_GRP	Yes/No	4	458	461	Is a default group assigned?
RACL_UTK_DFT_SECL	Yes/No	4	463	466	Is a default security label assigned?
RACL_PHASE	Char	20	468	487	Phase of this RACF command. Valid values are "LOCAL ISSUANCE", "TARGET PROCESSING", and "TARGET RESPONSE".
RACL_ISSUE_NODE	Char	8	489	496	Node that originated the command.
RACL_ISSUE_ID	Char	8	498	505	User ID that originated the command.
RACL_SOURCE_ID	Char	8	507	514	User ID for the association. From the ID keyword.
RACL_TGT_NODE	Char	8	516	523	Node that is the destination of the command.
RACL_TGT_ID	Char	8	525	532	User ID that is the destination of the command.
RACL_TGT_AUTH_ID	Char	8	534	541	User ID under whose authority the association is established.
RACL_SOURCE_SMFID	Char	4	543	546	SMF system identifier of the system that originated the command.
RACL_SOURCE_TIME	Char	8	548	555	Time that the command originated.
RACL_SOURCE_DATE	Char	10	557	566	Date that the command originated.

Table 125. Format of the RACLINK command record extension (event code 59) (continued)

Field name	Type	Length	Position		Comments
			Start	End	
RACL_PWD_STATUS	Char	8	568	575	<p>Status of the password sent with the command. Valid values are:</p> <p>SUPPLIED A password was supplied on a DEFINE command. This value occurs only for a LOCAL ISSUANCE phase record or for a TARGET PROCESSING phase when the event number is 3.</p> <p>VALID The password that was supplied on a DEFINE command is correct. This value occurs only for TARGET PROCESSING and TARGET RESPONSE phase records.</p> <p>NOTVALID The password that was supplied on a DEFINE command is not correct. This value occurs only for a TARGET PROCESSING phase record.</p> <p>EXPIRED The password that was supplied on a DEFINE command is expired. This value occurs for a TARGET PROCESSING only.</p> <p>REVOKED The target user ID on the DEFINE command is revoked. This value occurs for a TARGET PROCESSING phase only.</p> <p>NONE No password was supplied for the DEFINE command. This value can occur for any phase record.</p> <p>A blank value indicates that an UNDEFINE or APPROVE command was issued. Neither of these commands have passwords.</p>
RACL_ASSOC_STATUS	Char	8	577	584	Status of the association. Valid values are "PENDING", "ESTAB", and "DELETED".
RACL_SPECIFIED	Char	1024	586	1609	The keywords specified.
RACL_UTK_NETW	Char	8	1611	1618	The port of entry network name.
RACL_X500_SUBJECT	Char	255	1620	1874	Subject's name associated with this event.
RACL_X500_ISSUER	Char	255	1876	2130	Issuer's name associated with this event.
RACL_SERV_POENAME	Char	64	2132	2195	SERVAUTH resource or profile name.
RACL_CTX_USER	Char	510	2197	2706	Authenticated user name.
RACL_CTX_REG	Char	255	2708	2962	Authenticated user registry name.
RACL_CTX_HOST	Char	128	2964	3091	Authenticated user host name.
RACL_CTX_MECH	Char	16	3093	3108	Authenticated user authentication mechanism object identifier (OID).
RACL_IDID_USER	Char	985	3110	4094	Authenticated distributed user name.
RACL_IDID_REG	Char	1021	4096	5116	Authenticated distributed user registry name.

Note: Records created for user IDs that are revoked have no UTOKEN information.

<i>Table 126. Event qualifiers for RACLINK command records</i>		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Command successful.
INSAUTH	01	Insufficient authority (local issuance only).
-----	02	Reserved for IBM's use.
ALRDYDEF	03	Association already defined.
ALRDYAPP	04	Association already approved.
NOMATCH	05	Association does not match.
NOTEXIST	06	Association does not exist.
INVPSWD	07	Invalid password.

The IPCCHK record extension

Table 127 on page 292 describes the format of a record that is created by checking access to an IPC.

The event qualifiers that can be associated with a check IPC event are shown in Table 128 on page 294.

<i>Table 127. Format of the IPCCHK record extension (event code 60)</i>					
Field name	Type	Length	Position		Comments
			Start	End	
ICLK_CLASS	Char	8	282	289	Class name.
ICLK_USER_NAME	Char	20	291	310	The name associated with the user ID.
ICLK_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
ICLK_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
ICLK_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
ICLK_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
ICLK_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
ICLK_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
ICLK_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
ICLK_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
ICLK_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
ICLK_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
ICLK_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
ICLK_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
ICLK_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
ICLK_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
ICLK_UTK_SECL	Char	8	386	393	The security label of the user.
ICLK_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
ICLK_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
ICLK_UTK_SNODE	Char	8	413	420	The submitting node.
ICLK_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
ICLK_UTK_SPOE	Char	8	431	438	The port of entry.

Table 127. Format of the IPCCHK record extension (event code 60) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
ICLK_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ICLK_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
ICLK_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
ICLK_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
ICLK_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
ICLK_APPC_LINK	Char	16	477	492	Key to link together APPC records.
ICLK_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see z/OS Security Server RACF Callable Services .
ICLK_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
ICLK_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
ICLK_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
ICLK_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
ICLK_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
ICLK_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
ICLK_KEY_OWN_UID	Integer	10	572	581	The owner z/OS UNIX user identifier (UID) associated with the key.
ICLK_KEY_OWN_GID	Integer	10	583	592	The owner z/OS UNIX group identifier (GID) associated with the key.
ICLK_REQUEST_READ	Yes/No	4	594	597	Did the requested access include read?
ICLK_REQUEST_WRITE	Yes/No	4	599	602	Did the requested access include write?
ICLK_REQUEST_EXEC	Yes/No	4	604	607	Did the requested access include execute?
ICLK_RESERVED_01	Yes/No	4	609	612	Reserved for IBM's use.
ICLK_ACCESS_TYPE	Char	8	614	621	What bits were used in granting the access? Valid values are "OWNER", "GROUP", "NO", and "OTHER".
ICLK_ALLOWED_READ	Yes/No	4	623	626	Was read access allowed?
ICLK_ALLOWED_WRITE	Yes/No	4	628	631	Was write access allowed?
ICLK_RESERVED_02	Yes/No	4	633	636	Reserved for IBM's use.
ICLK_KEY	Char	8	638	645	The key of the IPC resource.
ICLK_ID	Integer	10	647	656	The unique decimal identifier of the IPC resource.
ICLK_CREATOR_UID	Integer	10	658	667	The z/OS UNIX user identifier (UID) of the creator.
ICLK_CREATOR_GID	Integer	10	669	678	The z/OS UNIX group identifier (GID) of the creator.
ICLK_DFLT_PROCESS	Yes/No	4	680	683	Default z/OS UNIX security environment in effect.
ICLK_UTK_NETW	Char	8	685	692	The port of entry network name.
ICLK_X500_SUBJECT	Char	255	694	948	Subject's name associated with this event.
ICLK_X500_ISSUER	Char	255	950	1204	Issuer's name associated with this event.
ICLK_SECL	Char	8	1206	1213	Security label of the resource.
ICLK_SERV_POENAME	Char	64	1215	1278	SERVAUTH resource or profile name.
ICLK_CTX_USER	Char	510	1280	1789	Authenticated user name.
ICLK_CTX_REG	Char	255	1791	2045	Authenticated user registry name.

Table 127. Format of the IPCCHK record extension (event code 60) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
ICLK_CTX_HOST	Char	128	2047	2174	Authenticated user host name.
ICLK_CTX_MECH	Char	16	2176	2191	Authenticated user authentication mechanism object identifier (OID).
ICLK_IDID_USER	Char	985	2193	3177	Authenticated distributed user name.
ICLK_IDID_REG	Char	1021	3179	4199	Authenticated distributed user registry name.

Table 128. Event qualifiers for check IPC records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	Access allowed.
NOTAUTH	01	Not authorized to the resource.
INSSECL	02	Insufficient security label.

The IPCGET record extension

Table 129 on page 294 describes the format of a record that is created by creating an IPC.

The event qualifiers that can be associated with an IPCGET event are shown in Table 130 on page 296.

Table 129. Format of the IPCGET record extension (event code 61)					
Field name	Type	Length	Position		Comments
			Start	End	
IGET_CLASS	Char	8	282	289	Class name.
IGET_USER_NAME	Char	20	291	310	The name associated with the user ID.
IGET_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
IGET_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
IGET_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
IGET_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
IGET_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
IGET_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
IGET_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
IGET_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
IGET_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
IGET_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
IGET_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
IGET_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
IGET_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
IGET_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
IGET_UTK_SECL	Char	8	386	393	The security label of the user.
IGET_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
IGET_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.

Table 129. Format of the IPCGET record extension (event code 61) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
IGET_UTK_SNODE	Char	8	413	420	The submitting node.
IGET_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
IGET_UTK_SPOE	Char	8	431	438	The port of entry.
IGET_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
IGET_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
IGET_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
IGET_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
IGET_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
IGET_APPC_LINK	Char	16	477	492	Key to link together APPC records.
IGET_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see z/OS Security Server RACF Callable Services .
IGET_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
IGET_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
IGET_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
IGET_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
IGET_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
IGET_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
IGET_KEY_OWN_UID	Integer	10	572	581	The owner z/OS UNIX user identifier (UID) associated with the key.
IGET_KEY_OWN_GID	Integer	10	583	592	The owner z/OS UNIX group identifier (GID) associated with the key.
IGET_RESERVED_01	Yes/No	4	594	597	Reserved for IBM's use.
IGET_RESERVED_02	Yes/No	4	599	602	Reserved for IBM's use.
IGET_RESERVED_03	Yes/No	4	604	607	Reserved for IBM's use.
IGET_REQ_OWN_READ	Yes/No	4	609	612	Was the owner READ bit requested on for this file?
IGET_REQ_OWN_WRITE	Yes/No	4	614	617	Was the owner WRITE bit requested on for this file?
IGET_REQ_OWN_EXEC	Yes/No	4	619	622	Was the owner EXECUTE bit requested on for this file?
IGET_REQ_GRP_READ	Yes/No	4	624	627	Was the group READ bit requested on for this file?
IGET_REQ_GRP_WRITE	Yes/No	4	629	632	Was the group WRITE bit requested on for this file?
IGET_REQ_GRP_EXEC	Yes/No	4	634	637	Was the group EXECUTE bit requested on for this file?
IGET_REQ_OTH_READ	Yes/No	4	639	642	Was the other READ bit requested on for this file?
IGET_REQ_OTH_WRITE	Yes/No	4	644	647	Was the other WRITE bit requested on for this file?
IGET_REQ_OTH_EXEC	Yes/No	4	649	652	Was the other EXECUTE bit requested on for this file?
IGET_KEY	Char	8	654	661	The key of the IPC resource.
IGET_ID	Integer	10	663	672	The unique decimal identifier of the IPC resource.
IGET_CREATOR_UID	Integer	10	674	683	The z/OS UNIX user identifier (UID) of the creator.
IGET_CREATOR_GID	Integer	10	685	694	The z/OS UNIX group identifier (GID) of the creator.
IGET_DFLT_PROCESS	Yes/No	4	696	699	Default z/OS UNIX security environment in effect.
IGET_UTK_NETW	Char	8	701	708	The port of entry network name.

Table 129. Format of the IPCGET record extension (event code 61) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
IGET_X500_SUBJECT	Char	255	710	964	Subject's name associated with this event.
IGET_X500_ISSUER	Char	255	966	1220	Issuer's name associated with this event.
IGET_SECL	Char	8	1222	1229	Security label of the resource.
IGET_SERV_POENAME	Char	64	1231	1294	SERVAUTH resource or profile name.
IGET_CTX_USER	Char	510	1296	1805	Authenticated user name.
IGET_CTX_REG	Char	255	1807	2061	Authenticated user registry name.
IGET_CTX_HOST	Char	128	2063	2190	Authenticated user host name.
IGET_CTX_MECH	Char	16	2192	2207	Authenticated user authentication mechanism object identifier (OID).
IGET_IDID_USER	Char	985	2209	3193	Authenticated distributed user name.
IGET_IDID_REG	Char	1021	3195	4215	Authenticated distributed user registry name.

Table 130. Event qualifiers for IPCGET records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Access allowed.
INSSECL	02	Insufficient security label.

The IPCCTL record extension

Table 131 on page 296 describes the format of a record that is created by the IPCCTL function.

The event qualifiers that can be associated with an IPCCTL event are shown in Table 132 on page 299.

Table 131. Format of the IPCCTL record extension (event code 62)					
Field name	Type	Length	Position		Comments
			Start	End	
ICTL_CLASS	Char	8	282	289	Class name.
ICTL_USER_NAME	Char	20	291	310	The name associated with the user ID.
ICTL_UTK_ENCR	Yes/No	4	312	315	Is the UTKEN associated with this user encrypted?
ICTL_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
ICTL_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
ICTL_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
ICTL_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
ICTL_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
ICTL_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
ICTL_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
ICTL_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
ICTL_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
ICTL_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
ICTL_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?

Table 131. Format of the IPCCTL record extension (event code 62) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
ICTL_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
ICTL_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
ICTL_UTK_SECL	Char	8	386	393	The security label of the user.
ICTL_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
ICTL_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
ICTL_UTK_SNODE	Char	8	413	420	The submitting node.
ICTL_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
ICTL_UTK_SPOE	Char	8	431	438	The port of entry.
ICTL_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ICTL_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
ICTL_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
ICTL_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
ICTL_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
ICTL_APPC_LINK	Char	16	477	492	Key to link together APPC records.
ICTL_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see z/OS Security Server RACF Callable Services .
ICTL_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
ICTL_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
ICTL_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
ICTL_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
ICTL_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
ICTL_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
ICTL_KEY_OWN_UID	Integer	10	572	581	The owner z/OS UNIX user identifier (UID) associated with the key
ICTL_KEY_OWN_GID	Integer	10	583	592	The owner z/OS UNIX group identifier (GID) associated with the key.
ICTL_UID	Integer	10	594	603	The owner z/OS UNIX user identifier (UID) input parameter.
ICTL_GID	Integer	10	605	614	The owner z/OS UNIX group identifier (GID) input parameter.
ICTL_RESERVED_01	Yes/No	4	616	619	Reserved for IBM's use.
ICTL_RESERVED_02	Yes/No	4	621	624	Reserved for IBM's use.
ICTL_RESERVED_03	Yes/No	4	626	629	Reserved for IBM's use.
ICTL_OLD_OWN_READ	Yes/No	4	631	634	Was the owner READ bit on for this file?
ICTL_OLD_OWN_WRITE	Yes/No	4	636	639	Was the owner WRITE bit on for this file?
ICTL_OLD_OWN_EXEC	Yes/No	4	641	644	Was the owner EXECUTE bit on for this file?
ICTL_OLD_GRP_READ	Yes/No	4	646	649	Was the group READ bit on for this file?
ICTL_OLD_GRP_WRITE	Yes/No	4	651	654	Was the group WRITE bit on for this file?
ICTL_OLD_GRP_EXEC	Yes/No	4	656	659	Was the group EXECUTE bit on for this file?

Table 131. Format of the IPCCTL record extension (event code 62) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
ICTL_OLD_OTH_READ	Yes/No	4	661	664	Was the other READ bit on for this file?
ICTL_OLD_OTH_WRITE	Yes/No	4	666	669	Was the other WRITE bit on for this file?
ICTL_OLD_OTH_EXEC	Yes/No	4	671	674	Was the other EXECUTE bit on for this file?
ICTL_RESERVED_04	Yes/No	4	676	679	Reserved for IBM's use.
ICTL_RESERVED_05	Yes/No	4	681	684	Reserved for IBM's use.
ICTL_RESERVED_06	Yes/No	4	686	689	Reserved for IBM's use.
ICTL_NEW_OWN_READ	Yes/No	4	691	694	Is the owner READ bit on for this file?
ICTL_NEW_OWN_WRITE	Yes/No	4	696	699	Is the owner WRITE bit on for this file?
ICTL_NEW_OWN_EXEC	Yes/No	4	701	704	Is the owner EXECUTE bit on for this file?
ICTL_NEW_GRP_READ	Yes/No	4	706	709	Is the group READ bit on for this file?
ICTL_NEW_GRP_WRITE	Yes/No	4	711	714	Is the group WRITE bit on for this file?
ICTL_NEW_GRP_EXEC	Yes/No	4	716	719	Is the group EXECUTE bit on for this file?
ICTL_NEW_OTH_READ	Yes/No	4	721	724	Is the other READ bit on for this file?
ICTL_NEW_OTH_WRITE	Yes/No	4	726	729	Is the other WRITE bit on for this file?
ICTL_NEW_OTH_EXEC	Yes/No	4	731	734	Is the other EXECUTE bit on for this file?
ICTL_SERVICE_CODE	Char	11	736	746	The service that was being processed.
ICTL_RESERVED_07	Yes/No	4	748	751	Reserved for IBM's use.
ICTL_RESERVED_08	Yes/No	4	753	756	Reserved for IBM's use.
ICTL_RESERVED_09	Yes/No	4	758	761	Reserved for IBM's use.
ICTL_REQ_OWN_READ	Yes/No	4	763	766	Was the owner READ bit requested on for this file?
ICTL_REQ_OWN_WRITE	Yes/No	4	768	771	Was the owner WRITE bit requested on for this file?
ICTL_REQ_OWN_EXEC	Yes/No	4	773	776	Was the owner EXECUTE bit requested on for this file?
ICTL_REQ_GRP_READ	Yes/No	4	778	781	Was the group READ bit requested on for this file?
ICTL_REQ_GRP_WRITE	Yes/No	4	783	786	Was the group WRITE bit requested on for this file?
ICTL_REQ_GRP_EXEC	Yes/No	4	788	791	Was the group EXECUTE bit requested on for this file?
ICTL_REQ_OTH_READ	Yes/No	4	793	796	Was the other READ bit requested on for this file?
ICTL_REQ_OTH_WRITE	Yes/No	4	798	801	Was the other WRITE bit requested on for this file?
ICTL_REQ_OTH_EXEC	Yes/No	4	803	806	Was the other EXECUTE bit requested on for this file?
ICTL_KEY	Char	8	808	815	The key of the IPC resource.
ICTL_ID	Integer	10	817	826	The unique decimal identifier of the IPC resource.
ICTL_CREATOR_UID	Integer	10	828	837	The z/OS UNIX user identifier (UID) of the creator.
ICTL_CREATOR_GID	Integer	10	839	848	The z/OS UNIX group identifier (GID) of the creator.
ICTL_DFLT_PROCESS	Yes/No	4	850	853	Default z/OS UNIX security environment in effect.
ICTL_UTK_NETW	Char	8	855	862	The port of entry network name.
ICTL_X500_SUBJECT	Char	255	864	1118	Subject's name associated with this event.
ICTL_X500_ISSUER	Char	255	1120	1374	Issuer's name associated with this event.
ICTL_SECL	Char	8	1376	1383	Security label of the resource.
ICTL_SERV_POENAME	Char	64	1385	1448	SERVAUTH resource or profile name.

Table 131. Format of the IPCCTL record extension (event code 62) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
ICTL_CTX_USER	Char	510	1450	1959	Authenticated user name.
ICTL_CTX_REG	Char	255	1961	2215	Authenticated user registry name.
ICTL_CTX_HOST	Char	128	2217	2344	Authenticated user host name.
ICTL_CTX_MECH	Char	16	2346	2361	Authenticated user authentication mechanism object identifier (OID).
ICTL_IDID_USER	Char	985	2363	3347	Authenticated distributed user name.
ICTL_IDID_REG	Char	1021	3349	4369	Authenticated distributed user registry name.

Table 132. Event qualifiers for IPCCTL records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Access allowed.
NOTAUTH	01	Not authorized to the resource.
INSSECL	02	Insufficient security label.

The SETGROUP record extension

Table 133 on page 299 describes the format of a record that is created by checking the owner of a file.

The event qualifiers that can be associated with the SETGROUP function are shown in Table 134 on page 301.

Table 133. Format of the SETGROUP record extension (event code 63)					
Field name	Type	Length	Position		Comments
			Start	End	
SETG_CLASS	Char	8	282	289	Class name.
SETG_USER_NAME	Char	20	291	310	The name associated with the user ID.
SETG_UTK_ENCR	Yes/No	4	312	315	Is the UTKEN associated with this user encrypted?
SETG_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
SETG_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
SETG_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
SETG_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
SETG_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
SETG_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
SETG_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
SETG_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
SETG_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
SETG_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
SETG_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
SETG_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
SETG_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
SETG_UTK_SECL	Char	8	386	393	The security label of the user.

Table 133. Format of the SETGROUP record extension (event code 63) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
SETG_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
SETG_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
SETG_UTK_SNODE	Char	8	413	420	The submitting node.
SETG_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
SETG_UTK_SPOE	Char	8	431	438	The port of entry.
SETG_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
SETG_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
SETG_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
SETG_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
SETG_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
SETG_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
SETG_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see z/OS Security Server RACF Callable Services .
SETG_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
SETG_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
SETG_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
SETG_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
SETG_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
SETG_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
SETG_DCE_LINK	Char	16	572	587	Link to connect DCE records that originate from a single DCE request.
SETG_AUTH_TYPE	Char	13	589	601	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
SETG_DFLT_PROCESS	Yes/No	4	603	606	Default z/OS UNIX security environment in effect.
SETG_UTK_NETW	Char	8	608	615	The port of entry network name.
SETG_X500_SUBJECT	Char	255	617	871	Subject's name associated with this event.
SETG_X500_ISSUER	Char	255	873	1127	Issuer's name associated with this event.
SETG_SERV_POENAME	Char	64	1129	1192	SERVAUTH resource or profile name.
SETG_CTX_USER	Char	510	1194	1703	Authenticated user name.
SETG_CTX_REG	Char	255	1705	1959	Authenticated user registry name.
SETG_CTX_HOST	Char	128	1961	2088	Authenticated user host name.
SETG_CTX_MECH	Char	16	2090	2105	Authenticated user authentication mechanism object identifier (OID).
SETG_IDID_USER	Char	985	2107	3091	Authenticated distributed user name.
SETG_IDID_REG	Char	1021	3093	4113	Authenticated distributed user registry name.

Table 134. Event qualifiers for SETGROUP records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	Process successfully initialized.
NOTAUTH	01	User does not have superuser authority.

The CKOWN2 record extension

Table 135 on page 301 describes the format of a record that is created by checking the owner of a file.

The event qualifiers that can be associated with checking a file's owner are shown in [Table 136 on page 303](#).

Table 135. Format of the CKOWN2 record extension (event code 64)

Field name	Type	Length	Position		Comments
			Start	End	
CKO2_CLASS	Char	8	282	289	Class name.
CKO2_USER_NAME	Char	20	291	310	The name associated with the user ID.
CKO2_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
CKO2_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
CKO2_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
CKO2_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
CKO2_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
CKO2_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
CKO2_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
CKO2_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
CKO2_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
CKO2_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
CKO2_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
CKO2_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
CKO2_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
CKO2_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
CKO2_UTK_SECL	Char	8	386	393	The security label of the user.
CKO2_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
CKO2_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
CKO2_UTK_SNODE	Char	8	413	420	The submitting node.
CKO2_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
CKO2_UTK_SPOE	Char	8	431	438	The port of entry.
CKO2_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
CKO2_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
CKO2_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
CKO2_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
CKO2_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?

Table 135. Format of the CKOWN2 record extension (event code 64) (continued)

Field name	Type	Length	Position		Comments
			Start	End	
CKO2_APPC_LINK	Char	16	477	492	Key to link together APPC records.
CKO2_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see z/OS Security Server RACF Callable Services .
CKO2_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
CKO2_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
CKO2_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
CKO2_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
CKO2_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
CKO2_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
CKO2_PATH_NAME	Char	1023	572	1594	The requested path name.
CKO2_FILE1_ID	Char	32	1596	1627	First file ID.
CKO2_FILE1_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the first file.
CKO2_FILE1_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the first file.
CKO2_FILE2_ID	Char	32	1651	1682	Second requested file ID.
CKO2_FILE2_OWN_UID	Integer	10	1684	1693	z/OS UNIX user identifier (UID) of the owner of the second file.
CKO2_FILE2_OWN_GID	Integer	10	1695	1704	z/OS UNIX group identifier (GID) of the owner of the second file.
CKO2_DCE_LINK	Char	16	1706	1721	Link to connect DCE records that originate from a single DCE request.
CKO2_AUTH_TYPE	Char	13	1723	1735	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
CKO2_DFLT_PROCESS	Yes/No	4	1737	1740	Default z/OS UNIX security environment in effect.
CKO2_UTK_NETW	Char	8	1742	1749	The port of entry network name.
CKO2_X500_SUBJECT	Char	255	1751	2005	Subject's name associated with this event.
CKO2_X500_ISSUER	Char	255	2007	2261	Issuer's name associated with this event.
CKO2_SECL	Char	8	2263	2270	Security label of the resource.
CKO2_SERV_POENAME	Char	64	2272	2335	SERVAUTH resource or profile name.
CKO2_CTX_USER	Char	510	2337	2846	Authenticated user name.
CKO2_CTX_REG	Char	255	2848	3102	Authenticated user registry name.
CKO2_CTX_HOST	Char	128	3104	3231	Authenticated user host name.
CKO2_CTX_MECH	Char	16	3233	3248	Authenticated user authentication mechanism object identifier (OID).
CKO2_IDID_USER	Char	985	3250	4234	Authenticated distributed user name.
CKO2_IDID_REG	Char	1021	4236	5256	Authenticated distributed user registry name.

Table 136. Event qualifiers for CKOWN2 records

Event qualifier	Event qualifier number	Event description
OWNER	00	Access allowed.
NOTOWNER	01	The user is not the owner.
INSSECL	02	Insufficient security label.

The access rights record extension

Table 137 on page 303 describes the format of a record that is created when access rights are passed.

The event qualifier that can be associated with access rights records are shown in Table 138 on page 304.

Table 137. Format of the access rights record extension (event code 65)

Field name	Type	Length	Position		Comments
			Start	End	
ACCR_CLASS	Char	8	282	289	Class name.
ACCR_USER_NAME	Char	20	291	310	The name associated with the user ID
ACCR_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
ACCR_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
ACCR_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
ACCR_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
ACCR_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
ACCR_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
ACCR_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
ACCR_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
ACCR_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
ACCR_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
ACCR_UTK_SESSTYPE	Char	8	362	369	The session type of this session
ACCR_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
ACCR_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
ACCR_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
ACCR_UTK_SECL	Char	8	386	393	The security label of the user
ACCR_UTK_EXECNODE	Char	8	395	402	The execution node of the work
ACCR_UTK_SUSER_ID	Char	8	404	411	The submitting user ID
ACCR_UTK_SNODE	Char	8	413	420	The submitting node
ACCR_UTK_SGRP_ID	Char	8	422	429	The submitting group name
ACCR_UTK_SPOE	Char	8	431	438	The port of entry
ACCR_UTK_SPCLASS	Char	8	440	447	Class of the POE Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT"
ACCR_UTK_USER_ID	Char	8	449	456	User ID associated with the record
ACCR_UTK_GRP_ID	Char	8	458	465	Group name associated with the record
ACCR_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
ACCR_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?

Table 137. Format of the access rights record extension (event code 65) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
ACCR_APPC_LINK	Char	16	477	492	Key to link together APPC records
ACCR_AUDIT_CODE	Char	11	494	504	Audit function code
ACCR_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID)
ACCR_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
ACCR_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
ACCR_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
ACCR_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
ACCR_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
ACCR_PATH_NAME	Char	1023	572	1594	The requested path name
ACCR_FILE1_ID	Char	32	1596	1627	File ID
ACCR_DFLT_PROCESS	Yes/No	4	1629	1632	Default z/OS UNIX security environment in effect.
ACCR_UTK_NETW	Char	8	1634	1641	The port of entry network name.
ACCR_X500_SUBJECT	Char	255	1643	1897	Subject's name associated with this event.
ACCR_X500_ISSUER	Char	255	1899	2153	Issuer's name associated with this event.
ACCR_SERV_POENAME	Char	64	2155	2218	SERVAUTH resource or profile name.
ACCR_CTX_USER	Char	510	2220	2729	Authenticated user name.
ACCR_CTX_REG	Char	255	2731	2985	Authenticated user registry name.
ACCR_CTX_HOST	Char	128	2987	3114	Authenticated user host name.
ACCR_CTX_MECH	Char	16	3116	3131	Authenticated user authentication mechanism object identifier (OID).
ACCR_IDID_USER	Char	985	3133	4117	Authenticated distributed user name.
ACCR_IDID_REG	Char	1021	4119	5139	Authenticated distributed user registry name.

Table 138. Event qualifiers for access rights records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Access rights are passed. There are no failure cases for this event.

The RACDCERT command record extension

Table 139 on page 304 describes the format of a record that is created by the RACDCERT command.

The event qualifiers that can be associated with the RACDCERT command are shown in Table 140 on page 306.

Table 139. Format of the RACDCERT command extension (event code 66)					
Field name	Type	Length	Position		Comments
			Start	End	
RACD_USER_NAME	Char	20	282	301	The name associated with the user ID.
RACD_UTK_ENCR	Yes/No	4	303	306	Is the UTKEN associated with this user encrypted?
RACD_UTK_PRE19	Yes/No	4	308	311	Is this a pre-1.9 token?
RACD_UTK_VERPROF	Yes/No	4	313	316	Is the VERIFYX propagation flag set?

Table 139. Format of the RACDCERT command extension (event code 66) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RACD_UTK_NJEUNUSR	Yes/No	4	318	321	Is this the NJE undefined user?
RACD_UTK_LOGUSR	Yes/No	4	323	326	Is UAUDIT specified for this user?
RACD_UTK_SPECIAL	Yes/No	4	328	331	Is this a SPECIAL user?
RACD_UTK_DEFAULT	Yes/No	4	333	336	Is this a default token?
RACD_UTK_UNKNUSR	Yes/No	4	338	341	Is this an undefined user?
RACD_UTK_ERROR	Yes/No	4	343	346	Is this user token in error?
RACD_UTK_TRUSTED	Yes/No	4	348	351	Is this user a part of the trusted computing base (TCG)?
RACD_UTK_SESSTYPE	Char	8	353	360	The session type of this session.
RACD_UTK_SURROGAT	Yes/No	4	362	365	Is this a surrogate user?
RACD_UTK_REMOTE	Yes/No	4	367	370	Is this a remote job?
RACD_UTK_PRIV	Yes/No	4	372	375	Is this a privileged user ID?
RACD_UTK_SECL	Char	8	377	384	The security label of the user.
RACD_UTK_EXECNODE	Char	8	386	393	The execution node of the work.
RACD_UTK_SUSER_ID	Char	8	395	402	The submitting user ID.
RACD_UTK_SNODE	Char	8	404	411	The submitting node.
RACD_UTK_SGRP_ID	Char	8	413	420	The submitting group name.
RACD_UTK_SPOE	Char	8	422	429	The port of entry.
RACD_UTK_SPCCLASS	Char	8	431	438	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT"
RACD_UTK_USER_ID	Char	8	440	447	User ID associated with the record.
RACD_UTK_GRP_ID	Char	8	449	456	Group name associated with the record.
RACD_UTK_DFT_GRP	Yes/No	4	458	461	Is a default group assigned?
RACD_UTK_DFT_SECL	Yes/No	4	463	466	Is a default security label assigned?
RACD_SERIAL_NUMBER	Char	255	468	722	Certificate serial number.
RACD_ISSUERS_DN	Char	255	724	978	Certificate issuer's distinguished name.
RACD_CERT_DS	Char	44	980	1023	Data set name containing the certificate.
RACD_SPECIFIED	Char	1024	1025	2048	The keywords specified.
RACD_UTK_NETW	Char	8	2050	2057	The port of entry network name.
RACD_X500_SUBJECT	Char	255	2059	2313	Subject's name associated with this event.
RACD_X500_ISSUER	Char	255	2315	2569	Issuer's name associated with this event.
RACD_SERV_POENAME	Char	64	2571	2634	SERVAUTH resource or profile name.
RACD_CTX_USER	Char	510	2636	3145	Authenticated user name.
RACD_CTX_REG	Char	255	3147	3401	Authenticated user registry name.
RACD_CTX_HOST	Char	128	3403	3530	Authenticated user host name.
RACD_CTX_MECH	Char	16	3532	3547	Authenticated user authentication mechanism object identifier (OID).
RACD_PKDS_LABEL	Char	64	3549	3612	PKDS label.
RACD_TOKEN	Char	32	3614	3645	Token name.
RACD_IDID_USER	Char	985	3647	4631	Authenticated distributed user name.

Table 139. Format of the RACDCERT command extension (event code 66) (continued)

Field name	Type	Length	Position		Comments
			Start	End	
RACD_IDID_REG	Char	1021	4633	5653	Authenticated distributed user registry name.
RACD_CERT_FGRPRNT	Char	64	5655	5718	Certificate SHA256 fingerprint in printable hex

Table 140. Event qualifiers for RACDCERT command records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	Command successful.
INSAUTH	01	Insufficient authority.

The InitACEE record extension

Table 141 on page 306 describes the format of a record that is created by InitACEE.

The event qualifiers that can be associated with InitACEE records are shown in Table 142 on page 307.

Table 141. Format of the InitACEE record extension (event code 67)

Field name	Type	Length	Position		Comments
			Start	End	
INTA_USER_NAME	Char	20	282	301	The name associated with the user ID.
INTA_UTK_ENCR	Yes/No	4	303	306	Is the UTOKEN associated with this user encrypted?
INTA_UTK_PRE19	Yes/No	4	308	311	Is this a pre-1.9 token?
INTA_UTK_VERPROF	Yes/No	4	313	316	Is the VERIFYX propagation flag set?
INTA_UTK_NJEUNUSR	Yes/No	4	318	321	Is this the NJE undefined user?
INTA_UTK_LOGUSR	Yes/No	4	323	326	Is UAUDIT specified for this user?
INTA_UTK_SPECIAL	Yes/No	4	328	331	Is this a SPECIAL user?
INTA_UTK_DEFAULT	Yes/No	4	333	336	Is this a default token?
INTA_UTK_UNKNUSR	Yes/No	4	338	341	Is this an undefined user?
INTA_UTK_ERROR	Yes/No	4	343	346	Is this user token in error?
INTA_UTK_TRUSTED	Yes/No	4	348	351	Is this user a part of the trusted computing base (TCB)?
INTA_UTK_SESSTYPE	Char	8	353	360	The session type of this session.
INTA_UTK_SURROGAT	Yes/No	4	362	365	Is this a surrogate user?
INTA_UTK_REMOTE	Yes/No	4	367	370	Is this a remote job?
INTA_UTK_PRIV	Yes/No	4	372	375	Is this a privileged user ID?
INTA_UTK_SECL	Char	8	377	384	The security label of the user.
INTA_UTK_EXECNODE	Char	8	386	393	The execution node of the work.
INTA_UTK_SUSER_ID	Char	8	395	402	The submitting user ID.
INTA_UTK_SNODE	Char	8	404	411	The submitting node.
INTA_UTK_SGRP_ID	Char	8	413	420	The submitting group name.
INTA_UTK_SPOE	Char	8	422	429	The port of entry.
INTA_UTK_SPCLASS	Char	8	431	438	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT"

Table 141. Format of the InitACEE record extension (event code 67) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
INTA_UTK_USER_ID	Char	8	440	447	User ID associated with the record.
INTA_UTK_GRP_ID	Char	8	449	456	Group name associated with the record.
INTA_UTK_DFT_GRP	Yes/No	4	458	461	Is a default group assigned?
INTA_UTK_DFT_SECL	Yes/No	4	463	466	Is a default security label assigned?
INTA_SERIAL_NUMBER	Char	255	468	722	Certificate serial number.
INTA_ISSUERS_DN	Char	255	724	978	Certificate issuer's distinguished name.
INTA_UTK_NETW	Char	8	980	987	The port of entry network name.
INTA_X500_SUBJECT	Char	255	989	1243	Subject's name associated with this event.
INTA_X500_ISSUER	Char	255	1245	1499	Issuer's name associated with this event.
INTA_SERVSECL	Char	8	1501	1508	Security label of server.
INTA_SERV_POENAME	Char	64	1510	1573	SERVAUTH resource or profile name.
INTA_CTX_USER	Char	510	1575	2084	Authenticated user name.
INTA_CTX_REG	Char	255	2086	2340	Authenticated user registry name.
INTA_CTX_HOST	Char	128	2342	2469	Authenticated user host name.
INTA_CTX_MECH	Char	16	2471	2486	Authenticated user authentication mechanism object identifier (OID).
INTA_IDID_USER	Char	985	2488	3472	Authenticated distributed user name.
INTA_IDID_REG	Char	1021	3474	4494	Authenticated distributed user registry name.
INTA_CERT_FGRPRNT	Char	64	4496	4559	Certificate SHA256 fingerprint in printable hex.
INTA_IDT_USER	Char	8	4561	4568	User ID from specified ACEE for generate IDT function
INTA_APPL	Char	8	4570	4577	Application name specified to initACEE for generate IDT function
INTA_IDT_BUILD_RSNC	Char	8	4579	4586	IDT Build Reason Code
INTA_SERVICE_CODE	Char	8	4588	4595	Failing Service Identifier
INTA_SERVICE_RC	Char	8	4597	4604	Failing Service Return Code
INTA_SERVICE_RSNC	Char	8	4606	4613	Failing Service Reason Code

Table 142. Event qualifiers for InitACEE records		
Event qualifier	Event qualifier number	Event description
SUCCSREG	00	Successful certificate registration.
SUCCSDER	01	Successful certificate deregistration.
INSAUREG	02	Insufficient authority to register the certificate.
INSAUDER	03	Insufficient authority to unregister the certificate.
NOUSRFND	04	No user ID found for the certificate.
CERNTRS	05	The certificate is not trusted.
SUCCSRCA	06	Successful CERTAUTH certificate registration.
INSAURCA	07	Insufficient authority to register the CERTAUTH certificate.
SECLSRVM	08	Mismatch with server's security label.
CERTRESV	09	A SITE or CERTAUTH certificate was used to authenticate a user.

Table 142. Event qualifiers for InitACEE records (continued)

Event qualifier	Event qualifier number	Event description
DIDNOTDF	10	No RACF user ID found for distributed identity.
SUCCSIDT	11	Successful IDT generated from ACEE.
FAILIDT	12	Failed attempting to generate IDT from ACEE.

The Network Authentication Service record extension

Table 143 on page 308 describes the format of a record that is created by the Network Authentication Service.

The event qualifiers that can be associated with Network Authentication Service records are shown in Table 144 on page 308.

Table 143. Format of the Network Authentication Service record extension (event code 68)

Field name	Type	Length	Position		Comments
			Start	End	
KTKT_PRINCIPAL	Char	240	282	521	The Kerberos principal name.
KTKT_LOGIN_SOURCE	Char	22	523	544	The Kerberos login request source.
KTKT_KDC_STAT_CODE	Char	10	546	555	The Kerberos KDC status code.

Table 144. Event qualifiers for Network Authentication Service records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful grant of initial Kerberos ticket.
FAILURE	01	Unsuccessful grant of initial Kerberos ticket.

The RPKIGENC record extension

Table 145 on page 308 describes the format of a record that is created by RPKIGENC.

The event qualifiers that can be associated with RPKIGENC records are shown in Table 146 on page 310.

Table 145. Format of the RPKIGENC record extension (event code 69)

Field name	Type	Length	Position		Comments
			Start	End	
RPKG_LOGSTRING	Char	255	282	536	Logstring parameter.
RPKG_USER_NAME	Char	20	538	557	The name associated with the user ID.
RPKG_UTK_ENCR	Yes/No	4	559	562	Is the UTOKEN associated with this user encrypted?
RPKG_UTK_PRE19	Yes/No	4	564	567	Is this a pre-1.9 token?
RPKG_UTK_VERPROF	Yes/No	4	569	572	Is the VERIFYX propagation?
RPKG_UTK_NJEUNUSR	Yes/No	4	574	577	Is this the NJE undefined user?
RPKG_UTK_LOGUSR	Yes/No	4	579	582	Is UAUDIT specified for this user?
RPKG_UTK_SPECIAL	Yes/No	4	584	587	Is this a SPECIAL user?
RPKG_UTK_DEFAULT	Yes/No	4	589	592	Is this a default token?
RPKG_UTK_UNKNUSR	Yes/No	4	594	597	Is this an undefined user?
RPKG_UTK_ERROR	Yes/No	4	599	602	Is this user token in error?

Table 145. Format of the RPKIGENC record extension (event code 69) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RPKG_UTK_TRUSTED	Yes/No	4	604	607	Is this user a part of the TCB?
RPKG_UTK_SESSTYPE	Char	8	609	616	The session type of this session.
RPKG_UTK_SURROGAT	Yes/No	4	618	621	Is this a surrogate user?
RPKG_UTK_REMOTE	Yes/No	4	623	626	Is this a remote job?
RPKG_UTK_PRIV	Yes/No	4	628	631	Is this a privileged user ID?
RPKG_UTK_SECL	Char	8	633	640	The security label of the user.
RPKG_UTK_EXECNODE	Char	8	642	649	The execution node of the work.
RPKG_UTK_SUSER_ID	Char	8	651	658	The submitting user ID.
RPKG_UTK_SNODE	Char	8	660	667	The submitting node.
RPKG_UTK_SGRP_ID	Char	8	669	676	The submitting group name.
RPKG_UTK_SPOE	Char	8	678	685	The port of entry.
RPKG_UTK_SPCLASS	Char	8	687	694	Class of the POE.
RPKG_UTK_USER_ID	Char	8	696	703	User ID associated with the record.
RPKG_UTK_GRP_ID	Char	8	705	712	Group name associated with the record.
RPKG_UTK_DFT_GRP	Yes/No	4	714	717	Is a default group assigned?
RPKG_UTK_DFT_SECL	Yes/No	4	719	722	Is a default security label assigned?
RPKG_SERIAL_NUMBER	Char	255	724	978	Certificate serial number.
RPKG_ISSUERS_DN	Char	255	980	1234	Certificate issuer's distinguished name.
RPKG_UTK_NETW	Char	8	1236	1243	The port of entry network name.
RPKG_X500_SUBJECT	Char	255	1245	1499	Subject's name associated with this event.
RPKG_X500_ISSUER	Char	255	1501	1755	Issuer's name associated with this event.
RPKG_KEYUSAGE	Char	64	1757	1820	Requested certificate KeyUsage.
RPKG_NOTBEFOR_DATE	Char	10	1822	1831	Requested certificate NotBefore date.
RPKG_NOTAFTER_DATE	Char	10	1833	1842	Requested certificate NotAfter date.
RPKG_TARGET_USERID	Char	8	1844	1851	IRRSPX00 target user ID.
RPKG_TARGET_LABEL	Char	32	1853	1884	IRRSPX00 target label.
RPKG_SIGNWITH	Char	45	1886	1930	IRRSPX00 SignWith value.
RPKG_SUBJECTS_DN	Char	255	1932	2186	Certificate subject's distinguished name.
RPKG_ALT_IP	Char	64	2188	2251	Requested ALTNAME IP address.
RPKG_ALT_URI	Char	255	2253	2507	Requested ALTNAME URI.
RPKG_ALT_EMAIL	Char	100	2509	2608	Requested ALTNAME Email.
RPKG_ALT_DOMAIN	Char	100	2610	2709	Requested ALTNAME Domain.
RPKG_CERT_ID	Char	56	2711	2766	IRRSPX00 Certificate ID.
RPKG_HOSTID_MAP	Char	1024	2768	3791	HOSTID mappings extension data.
RPKG_REQUESTOR	Char	32	3793	3824	Requester's name.
RPKG_PASS_PHRASE	Yes/No	4	3826	3829	Requester specified a pass phrase.
RPKG_NOTIFY_EMAIL	Char	64	3831	3894	Email address for notification purposes.
RPKG_EXTKEYUSAGE	Char	255	3896	4150	Requested Extended KeyUsage.

Table 145. Format of the RPKIGENC record extension (event code 69) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RPKG_CERTPOLICIES	Char	32	4152	4183	Policies for certificate usage.
RPKG_AUTHINFOACC	Char	1024	4185	5208	AuthorityInfoAccess extension data.
RPKG_CRITICAL	Char	255	5210	5464	Extensions marked critical.
RPKG_SERV_POENAME	Char	64	5466	5529	SERVAUTH resource or profile name.
RPKG_ALT_OTHER	Char	1024	5531	6554	Requested ALTNAME OtherName
RPKG_CA_DOMAIN	Char	8	6556	6563	Domain name of target PKI Services instance
RPKG_CTX_USER	Char	510	6565	7074	Authenticated user name.
RPKG_CTX_REG	Char	255	7076	7330	Authenticated user registry name.
RPKG_CTX_HOST	Char	128	7332	7459	Authenticated user host name.
RPKG_CTX_MECH	Char	16	7461	7476	Authenticated user authentication mechanism object identifier (OID).
RPKG_KEY_SIZE	Char	4	7478	7481	Key size
RPKG_IDID_USER_UTF8	Char	246	7483	7728	Authenticated distributed user name in UTF-8.
RPKG_IDID_USER_EBCDIC	Char	738	7730	8467	Authenticated distributed user name in EBCDIC.
RPKG_IDID_REG_UTF8	Char	255	8469	8723	Authenticated distributed registry name in UTF-8.
RPKG_IDID_REG_EBCDIC	Char	765	8725	9489	Authenticated distributed registry name in EBCDIC.
RPKG_KEY_ALG	Char	10	9491	9500	Key algorithm.
RPKG_CUSTOM_EXT	Char	1024	9502	10525	Customized extension.
RPKG_RECORD_LINK	Char	32	10527	10558	Field to link audit records together.
RPKG_CERT_FGRPRNT	Char	64	10560	10623	Subject Certificate SHA256 fingerprint in printable hex value

Table 146. Event qualifiers for RPKIGENC records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful certificate GENCERT request.
INSAUTH	01	Unsuccessful certificate GENCERT request because of insufficient authority.
SUCCESSRQC	02	Successful certificate REQCERT request.
IAUTHRQC	03	Unsuccessful certificate REQCERT request because of insufficient authority.
SUCCESSGNR	04	Successful certificate GENRENEW request.
IAUTHGNR	05	Unsuccessful certificate GENRENEW request because of insufficient authority.
SUCCESSRQR	06	Successful certificate REQRENEW request.
IAUTHRQR	07	Unsuccessful certificate REQRENEW request because of insufficient authority.
SUCCESSPRG	08	Successful PREREGISTER request.
IAUTHPRG	09	Insufficient authority for PREREGISTER

The RPKIEXPT record extension

Table 147 on page 311 describes the format of a record that is created by RPKIEXPT.

The event qualifiers that can be associated with RPKIEXPT records are shown in [Table 148 on page 312](#).

Table 147. Format of the RPKIEXPT record extension (event code 70)					
Field name	Type	Length	Position		Comments
			Start	End	
RPKE_LOGSTRING	Char	255	282	536	Logstring parameter.
RPKE_USER_NAME	Char	20	538	557	The name associated with the user ID.
RPKE_UTK_ENCR	Yes/No	4	559	562	Is the UTKEN associated with this user encrypted?
RPKE_UTK_PRE19	Yes/No	4	564	567	Is this a pre-1.9 token?
RPKE_UTK_VERPROF	Yes/No	4	569	572	Is the VERIFYX propagation.
RPKE_UTK_NJEUNUSR	Yes/No	4	574	577	Is this the NJE undefined user?
RPKE_UTK_LOGUSR	Yes/No	4	579	582	Is UAUDIT specified for this user?
RPKE_UTK_SPECIAL	Yes/No	4	584	587	Is this a SPECIAL user?
RPKE_UTK_DEFAULT	Yes/No	4	589	592	Is this a default token?
RPKE_UTK_UNKNUSR	Yes/No	4	594	597	Is this an undefined user?
RPKE_UTK_ERROR	Yes/No	4	599	602	Is this user token in error?
RPKE_UTK_TRUSTED	Yes/No	4	604	607	Is this user a part of the TCB?
RPKE_UTK_SESSTYPE	Char	8	609	616	The session type of this session.
RPKE_UTK_SURROGAT	Yes/No	4	618	621	Is this a surrogate user?
RPKE_UTK_REMOTE	Yes/No	4	623	626	Is this a remote job?
RPKE_UTK_PRIV	Yes/No	4	628	631	Is this a privileged user ID?
RPKE_UTK_SECL	Char	8	633	640	The security label of the user.
RPKE_UTK_EXECNODE	Char	8	642	649	The execution node of the work.
RPKE_UTK_SUSER_ID	Char	8	651	658	The submitting user ID.
RPKE_UTK_SNODE	Char	8	660	667	The submitting node.
RPKE_UTK_SGRP_ID	Char	8	669	676	The submitting group name.
RPKE_UTK_SPOE	Char	8	678	685	The port of entry.
RPKE_UTK_SPCLASS	Char	8	687	694	Class of the POE.
RPKE_UTK_USER_ID	Char	8	696	703	User ID associated with the record.
RPKE_UTK_GRP_ID	Char	8	705	712	Group name associated with the record.
RPKE_UTK_DFT_GRP	Yes/No	4	714	717	Is a default group assigned?
RPKE_UTK_DFT_SECL	Yes/No	4	719	722	Is a default security label assigned?
RPKE_UTK_NETW	Char	8	724	731	The port of entry network name.
RPKE_X500_SUBJECT	Char	255	733	987	Subject's name associated with this event.
RPKE_X500_ISSUER	Char	255	989	1243	Issuer's name associated with this event.
RPKE_TARGET_USERID	Char	8	1245	1252	IRRSPX00 target user ID.
RPKE_TARGET_LABEL	Char	32	1254	1285	IRRSPX00 target label.
RPKE_CERT_ID	Char	56	1287	1342	IRRSPX00 certificate ID.
RPKE_PASS_PHRASE	Yes/No	4	1344	1347	Requestor specified a pass phrase.
RPKE_SERV_POENAME	Char	64	1349	1412	SERVAUTH resource or profile name.
RPKE_CA_DOMAIN	Char	8	1414	1421	Domain name of target PKI Services instance.
RPKE_CTX_USER	Char	510	1423	1932	Authenticated user name.

Table 147. Format of the RPKIEXPT record extension (event code 70) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RPKE_CTX_REG	Char	255	1934	2188	Authenticated user registry name.
RPKE_CTX_HOST	Char	128	2190	2317	Authenticated user host name.
RPKE_CTX_MECH	Char	16	2319	2334	Authenticated user authentication mechanism object identifier (OID).
RPKE_KEY_ID	Char	40	2336	2375	Hash of the public key generated by PKI Services.
RPKE_IDID_USER	Char	985	2377	3361	Authenticated distributed user name.
RPKE_IDID_REG	Char	1021	3363	4383	Authenticated distributed user registry name.
RPKG_CERT_FGRPRNT	Char	64	4385	4448	Subject Certificate SHA256 fingerprint in printable hex value

Table 148. Event qualifiers for RPKIEXPT records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful certificate EXPORT request.
INSAUTH	01	Unsuccessful certificate EXPORT request because of insufficient authority.
INCORPHR	02	Incorrect pass phrase specified for EXPORT.

The Policy Director Authorization Services record extension

Table 149 on page 312 describes the format of a record that is created for Policy Director Authorization Services.

The event qualifiers that can be associated with Policy Director Authorization Services records are shown in Table 150 on page 312.

Table 149. Format of Policy Director Authorization Services record extension (event code 71)					
Field name	Type	Length	Position		Comments
			Start	End	
PDAC_OBJECT	Char	4096	282	4377	The Policy Director Authorization Services protected object.
PDAC_REQ_PERMS	Char	1024	4379	5402	The requested Policy Director Authorization Services permissions.
PDAC_HOST_USERID	Char	8	5404	5411	The Policy Director Authorization Services principal user ID.
PDAC_PRINCIPAL	Char	36	5413	5448	The Policy Director Authorization Services principal ID string.
PDAC_QOP	Integer	10	5450	5459	The Policy Director Authorization Services quality of protection value.
PDAC_CRED_TYPE	Char	30	5461	5490	The Policy Director Authorization Services credential type. The valid types are: "UNAUTHENTICATED" and "AUTHENTICATED"

Table 150. Event qualifiers for Policy Director Authorization Services records

Event qualifier	Event qualifier number	Event description
AUTH	00	Authorized to access protected object.

Table 150. Event qualifiers for Policy Director Authorization Services records (continued)		
Event qualifier	Event qualifier number	Event description
UNAUTHW	01	Not authorized to access protected object but permitted because of warning mode.
INSTRAVW	02	Not authorized to access protected object because of insufficient traverse authority but permitted because of warning mode.
TODW	03	Not authorized to access protected object because of time-of-day check but permitted because of warning mode.
UNAUTH	04	Not authorized to access protected object.
INSTRAV	05	Not authorized to access protected object because of insufficient traverse authority.
TOD	06	Not authorized to access protected object because of time-of-day check.

The RPKIREAD record extension

Table 151 on page 313 describes the format of a record that is created by RPKIREAD.

The event qualifiers that can be associated with RPKIREAD records are shown in Table 152 on page 315.

Table 151. Format of the RPKIREAD record extension (event code 72)					
Field name	Type	Length	Position		Comments
			Start	End	
RPKR_APPL	Char	8	282	289	Logstring parameter.
RPKR_LOGSTRING	Char	255	291	545	Logstring parameter.
RPKR_USER_NAME	Char	20	547	566	The name associated with the user ID.
RPKR_UTK_ENCR	Yes/No	4	568	571	Is the UTOKEN associated with this user encrypted?
RPKR_UTK_PRE19	Yes/No	4	573	576	Is this a pre-1.9 token?
RPKR_UTK_VERPROF	Yes/No	4	578	581	Is the VERIFYX propagation?
RPKR_UTK_NJEUNUSR	Yes/No	4	583	586	Is this the NJE undefined user?
RPKR_UTK_LOGUSR	Yes/No	4	588	591	Is UAUDIT specified for this user?
RPKR_UTK_SPECIAL	Yes/No	4	593	596	Is this a SPECIAL user?
RPKR_UTK_DEFAULT	Yes/No	4	598	601	Is this a default token?
RPKR_UTK_UNKNUSR	Yes/No	4	603	606	Is this an undefined user?
RPKR_UTK_ERROR	Yes/No	4	608	611	Is this user token in error?
RPKR_UTK_TRUSTED	Yes/No	4	613	616	Is this user a part of the TCB?
RPKR_UTK_SESSTYPE	Char	8	618	625	The session type of this session.
RPKR_UTK_SURROGAT	Yes/No	4	627	630	Is this a surrogate user?
RPKR_UTK_REMOTE	Yes/No	4	632	635	Is this a remote job?
RPKR_UTK_PRIV	Yes/No	4	637	640	Is this a privileged user ID?
RPKR_UTK_SECL	Char	8	642	649	The security label of the user.
RPKR_UTK_EXECNODE	Char	8	651	658	The execution node of the work.
RPKR_UTK_SUSER_ID	Char	8	660	667	The submitting user ID.
RPKR_UTK_SNODE	Char	8	669	676	The submitting node.
RPKR_UTK_SGRP_ID	Char	8	678	685	The submitting group name.
RPKR_UTK_SPOE	Char	8	687	694	The port of entry.

Table 151. Format of the RPKIREAD record extension (event code 72) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RPKR_UTK_SPCLASS	Char	8	696	703	Class of the POE.
RPKR_UTK_USER_ID	Char	8	705	712	User ID associated with the record.
RPKR_UTK_GRP_ID	Char	8	714	721	Group name associated with the record.
RPKR_UTK_DFT_GRP	Yes/No	4	723	726	Is a default group assigned?
RPKR_UTK_DFT_SECL	Yes/No	4	728	731	Is a default security label assigned?
RPKR_SERIAL_NUMBER	Char	255	733	987	Certificate serial number.
RPKR_ISSUERS_DN	Char	255	989	1243	Certificate issuer's distinguished name.
RPKR_UTK_NETW	Char	8	1245	1252	The port of entry network name.
RPKR_X500_SUBJECT	Char	255	1254	1508	Subject's name associated with this event.
RPKR_X500_ISSUER	Char	255	1510	1764	Issuer's name associated with this event.
RPKR_KEYUSAGE	Char	64	1766	1829	Requested certificate KeyUsage.
RPKR_NOTBEFOR_DATE	Char	10	1831	1840	Requested certificate NotBefore date.
RPKR_NOTAFTER_DATE	Char	10	1842	1851	Requested certificate NotAfter date.
RPKR_SUBJECTS_DN	Char	255	1853	2107	Certificate subject's distinguished name.
RPKR_CERT_ID	Char	56	2109	2164	IRRSPX00 Certificate ID.
RPKR_REQUESTOR	Char	32	2166	2197	Requester's name.
RPKR_STATUS	Char	32	2199	2230	Requester certificate status.
RPKR_CREATION_DATE	Char	10	2232	2241	Requester certificate creation date (YYYY/MM/DD).
RPKR_LAST_MOD_DATE	Char	10	2243	2252	Requester certificate last modification date (YYYY/MM/DD).
RPKR_PREV_SERIAL	Char	255	2254	2508	Requester's previous serial number.
RPKR_NOTIFY_EMAIL	Char	64	2510	2573	Email address for notification purposes.
RPKR_EXTKEYUSAG	Char	255	2575	2829	Requested Extended KeyUsage.
RPKR_SERV_POENAME	Char	64	2831	2894	SERVAUTH resource or profile name.
RPKR_CA_DOMAIN	Char	8	2896	2903	Domain name of target PKI Services instance.
RPKR_CTX_USER	Char	510	2905	3414	Authenticated user name.
RPKR_CTX_REG	Char	255	3416	3670	Authenticated user registry name.
RPKR_CTX_HOST	Char	128	3672	3799	Authenticated user host name.
RPKR_CTX_MECH	Char	16	3801	3816	Authenticated user authentication mechanism object identifier (OID).
RPKR_KEY_ID	Char	40	3818	3857	Hash of the public key generated by PKI Services.
RPKR_IDID_USER	Char	985	3859	4843	Authenticated distributed user name.
RPKR_IDID_REG	Char	1021	4845	5865	Authenticated distributed user registry name.
RPKR_KEY_SIZE	Char	4	5867	5870	Key size
RPKR_KEY_ALG	Char	10	5872	5881	Key algorithm
RPKR_SIGN_ALG	Char	32	5883	5914	Signing algorithm of a certificate request or a certificate
RPKR_APPROVAL_REQ	Integer	2	5916	5917	Number of approvals required for the request
RPKR_APPROVAL_CNT	Integer	2	5919	5920	Count of approvals performed

Table 152. Event qualifiers for RPKIREAD records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful admin QUERY or DETAILS request.
INSAUTH	01	Unsuccessful certificate admin QUERY or DETAILS request because of insufficient authority.
SUCCSVFY	02	Successful certificate VERIFY request.
IAUTHVFY	03	Unsuccessful certificate VERIFY request because of insufficient authority.
INCORCRT	04	Incorrect VERIFY certificate, no record found for this certificate.

The RPKIUPDR record extension

Table 153 on page 315 describes the format of a record that is created by RPKIUPDR.

The event qualifiers that can be associated with RPKIUPDR records are shown in Table 154 on page 317.

Table 153. Format of the RPKIUPDR record extension (event code 73)

Field name	Type	Length	Position		Comments
			Start	End	
RPKU_LOGSTRING	Char	255	282	536	Logstring parameter.
RPKU_USER_NAME	Char	20	538	557	The name associated with the user ID.
RPKU_UTK_ENCR	Yes/No	4	559	562	Is the UTOKEN associated with this user encrypted?
RPKU_UTK_PRE19	Yes/No	4	564	567	Is this a pre-1.9 token?
RPKU_UTK_VERPROF	Yes/No	4	569	572	Is the VERIFYX propagation?
RPKU_UTK_NJEUNUSR	Yes/No	4	574	577	Is this the NJE undefined user?
RPKU_UTK_LOGUSR	Yes/No	4	579	582	Is UAUDIT specified for this user?
RPKU_UTK_SPECIAL	Yes/No	4	584	587	Is this a SPECIAL user?
RPKU_UTK_DEFAULT	Yes/No	4	589	592	Is this a default token?
RPKU_UTK_UNKNUSR	Yes/No	4	594	597	Is this an undefined user?
RPKU_UTK_ERROR	Yes/No	4	599	602	Is this user token in error?
RPKU_UTK_TRUSTED	Yes/No	4	604	607	Is this user a part of the TCB?
RPKU_UTK_SESTYPE	Char	8	609	616	The session type of this session.
RPKU_UTK_SURROGAT	Yes/No	4	618	621	Is this a surrogate user?
RPKU_UTK_REMOTE	Yes/No	4	623	626	Is this a remote job?
RPKU_UTK_PRIV	Yes/No	4	628	631	Is this a privileged user ID?
RPKU_UTK_SECL	Char	8	633	640	The security label of the user.
RPKU_UTK_EXECNODE	Char	8	642	649	The execution node of the work.
RPKU_UTK_SUSER_ID	Char	8	651	658	The submitting user ID.
RPKU_UTK_SNODE	Char	8	660	667	The submitting node.
RPKU_UTK_SGRP_ID	Char	8	669	676	The submitting group name.
RPKU_UTK_SPOE	Char	8	678	685	The port of entry.
RPKU_UTK_SPCLASS	Char	8	687	694	Class of the POE.
RPKU_UTK_USER_ID	Char	8	696	703	User ID associated with the record.
RPKU_UTK_GRP_ID	Char	8	705	712	Group name associated with the record.

Table 153. Format of the RPKIUPDR record extension (event code 73) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RPKU_UTK_DFT_GRP	Yes/No	4	714	717	Is a default group assigned?
RPKU_UTK_DFT_SECL	Yes/No	4	719	722	Is a default security label assigned?
RPKU_UTK_NETW	Char	8	724	731	The port of entry network name.
RPKU_X500_SUBJECT	Char	255	733	987	Subject's name associated with this event.
RPKU_X500_ISSUER	Char	255	989	1243	Issuer's name associated with this event.
RPKU_KEYUSAGE	Char	64	1245	1308	Requested certificate KeyUsage.
RPKU_NOTBEFOR_DATE	Char	10	1310	1319	Requested certificate NotBefore date.
RPKU_NOTAFTER_DATE	Char	10	1321	1330	Requested certificate NotAfter date.
RPKU_SUBJECTS_DN	Char	255	1332	1586	Certificate subject's distinguished name.
RPKU_ALT_IP	Char	64	1588	1651	Requested ALTNAME IP address.
RPKU_ALT_URI	Char	255	1653	1907	Requested ALTNAME URI.
RPKU_ALT_EMAIL	Char	100	1909	2008	Requested ALTNAME EMail.
RPKU_ALT_DOMAIN	Char	100	2010	2109	Requested ALTNAME Domain.
RPKU_CERT_ID	Char	56	2111	2166	IRRSPX00 Certificate ID.
RPKU_HOSTID_MAP	Char	1024	2168	3191	HOSTID mappings extension data.
RPKU_ACTION	Char	16	3193	3208	Action taken against certificate request.
RPKU_ACTION_COM	Char	64	3210	3273	Comment for the action on the certificate request.
RPKU_EXTKEYUSAGE	Char	255	3275	3529	Requested Extended KeyUsage.
RPKU_CERTPOLICIES	Char	32	3531	3562	Policies for certificate usage.
RPKU_AUTHINFOACC	Char	1024	3564	4587	AuthorityInfoAccess extension data.
RPKU_CRITICAL	Char	255	4589	4843	Extensions marked critical.
RPKU_SERV_POENAME	Char	64	4845	4908	SERVAUTH resource or profile name.
RPKU_ALT_OTHER	Char	1024	4910	5933	Requested ALTNAME OtherName.
RPKU_CA_DOMAIN	Char	8	5935	5942	Domain name of target PKI Services instance.
RPKU_CTX_USER	Char	510	5944	6453	Authenticated user name.
RPKU_CTX_REG	Char	255	6455	6709	Authenticated user registry name.
RPKU_CTX_HOST	Char	128	6711	6838	Authenticated user host name.
RPKU_CTX_MECH	Char	16	6840	6855	Authenticated user authentication mechanism object identifier (OID).
RPKU_IDID_USER_UTF8	Char	246	6857	7102	Authenticated distributed user name in UTF-8.
RPKU_IDID_USER_EBCDIC	Char	738	7104	7841	Authenticated distributed user name in EBCDIC.
RPKU_IDID_REG_UTF8	Char	255	7843	8097	Authenticated distributed registry name in UTF-8.
RPKU_IDID_REG_EBCDIC	Char	765	8099	8863	Authenticated distributed registry name in EBCDIC.
RPKU_CUSTOM_EXT	Char	1024	8865	9888	Customized extension.
RPKU_RECORD_LINK	Char	32	9890	9921	Field to link audit records together.

Table 154. Event qualifiers for RPKIUPDR records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful admin UPDATEREQ request.
INSAUTH	01	Unsuccessful admin UPDATEREQ request because of insufficient authority.

The RPKIUPDC record extension

Table 155 on page 317 describes the format of a record that is created by RPKIUPDC.

The event qualifiers that can be associated with RPKIUPDC records are shown in Table 156 on page 318.

Table 155. Format of the RPKIUPDC record extension (event code 74)

Field name	Type	Length	Position		Comments
			Start	End	
RPKC_LOGSTRING	Char	255	282	536	Logstring parameter.
RPKC_USER_NAME	Char	20	538	557	The name associated with the user ID.
RPKC_UTK_ENCR	Yes/No	4	559	562	Is the UTOKEN associated with this user encrypted?
RPKC_UTK_PRE19	Yes/No	4	564	567	Is this a pre-1.9 token?
RPKC_UTK_VERPROF	Yes/No	4	569	572	Is the VERIFYX propagation?
RPKC_UTK_NJEUNUSR	Yes/No	4	574	577	Is this the NJE undefined user?
RPKC_UTK_LOGUSR	Yes/No	4	579	582	Is UAUDIT specified for this user?
RPKC_UTK_SPECIAL	Yes/No	4	584	587	Is this a SPECIAL user?
RPKC_UTK_DEFAULT	Yes/No	4	589	592	Is this a default token?
RPKC_UTK_UNKNUSR	Yes/No	4	594	597	Is this an undefined user?
RPKC_UTK_ERROR	Yes/No	4	599	602	Is this user token in error?
RPKC_UTK_TRUSTED	Yes/No	4	604	607	Is this user a part of the TCB?
RPKC_UTK_SESTYPE	Char	8	609	616	The session type of this session.
RPKC_UTK_SURROGAT	Yes/No	4	618	621	Is this a surrogate user?
RPKC_UTK_REMOTE	Yes/No	4	623	626	Is this a remote job?
RPKC_UTK_PRIV	Yes/No	4	628	631	Is this a privileged user ID?
RPKC_UTK_SECL	Char	8	633	640	The security label of the user.
RPKC_UTK_EXECNODE	Char	8	642	649	The execution node of the work.
RPKC_UTK_SUSER_ID	Char	8	651	658	The submitting user ID.
RPKC_UTK_SNODE	Char	8	660	667	The submitting node.
RPKC_UTK_SGRP_ID	Char	8	669	676	The submitting group name.
RPKC_UTK_SPOE	Char	8	678	685	The port of entry.
RPKC_UTK_SPCLASS	Char	8	687	694	Class of the POE.
RPKC_UTK_USER_ID	Char	8	696	703	User ID associated with the record.
RPKC_UTK_GRP_ID	Char	8	705	712	Group name associated with the record.
RPKC_UTK_DFT_GRP	Yes/No	4	714	717	Is a default group assigned?
RPKC_UTK_DFT_SECL	Yes/No	4	719	722	Is a default security label assigned?
RPKC_SERIAL_NUMBER	Char	255	724	978	Certificate serial number.
RPKC_UTK_NETW	Char	8	980	987	The port of entry network name.

Table 155. Format of the RPKIUPDC record extension (event code 74) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RPKC_X500_SUBJECT	Char	255	989	1243	Subject's name associated with this event.
RPKC_X500_ISSUER	Char	255	1245	1499	Issuer's name associated with this event.
RPKC_ACTION	Char	16	1501	1516	Action taken against certificate request.
RPKC_ACTION_COM	Char	64	1518	1581	Comment for the certificate request.
RPKC_REVOKE_RSN	Char	32	1583	1614	Reason for certificate revocation.
RPKC_SERV_POENAME	Char	64	1616	1679	SERVAUTH resource or profile name.
RPKC_CA_DOMAIN	Char	8	1681	1688	Domain name of target PKI Services instance.
RPKC_CTX_USER	Char	510	1690	2199	Authenticated user name.
RPKC_CTX_REG	Char	255	2201	2455	Authenticated user registry name.
RPKC_CTX_HOST	Char	128	2457	2584	Authenticated user host name.
RPKC_CTX_MECH	Char	16	2586	2601	Authenticated user authentication mechanism object identifier (OID).
RPKC_REQUESTOR_EMAIL	Char	32	2603	2634	New email address of the requester.
RPKC_IDID_USER	Char	985	2636	3620	Authenticated distributed user name.
RPKC_IDID_REG	Char	1021	3622	4642	Authenticated distributed user registry name.
RPKC_CERT_FGRPRNT	Char	64	4644	4707	Subject Certificate SHA256 fingerprint in printable hex value

Table 156. Event qualifiers for RPKIUPDC records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful admin UPDATECERT request.
INSAUTH	01	Unsuccessful admin UPDATECERT request because of insufficient authority.
SUCCSRVK	02	Successful certificate REVOKE request.
IAUTHRVK	03	Unsuccessful certificate REVOKE request because of insufficient authority.

The SETFACL record extension

Table 157 on page 318 describes the format of a record that is created by adding, modifying, or deleting an access control list entry of a z/OS UNIX file.

The event qualifiers that can be associated with an access control list modification event are shown in Table 158 on page 320.

Table 157. Format of the SETFACL record extension (event code 75)					
Field name	Type	Length	Position		Comments
			Start	End	
SACL_CLASS	Char	8	282	289	Class name.
SACL_USER_NAME	Char	20	291	310	The name associated with the user ID.
SACL_UTK_ENCR	Yes/No	4	312	315	Is the UTKEN associated with this user encrypted?
SACL_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
SACL_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
SACL_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?

Table 157. Format of the SETFACL record extension (event code 75) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
SACL_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
SACL_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
SACL_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
SACL_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
SACL_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
SACL_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
SACL_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
SACL_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
SACL_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
SACL_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
SACL_UTK_SECL	Char	8	386	393	The security label of the user.
SACL_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
SACL_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
SACL_UTK_SNODE	Char	8	413	420	The submitting node.
SACL_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
SACL_UTK_SPOE	Char	8	431	438	The port of entry.
SACL_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
SACL_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
SACL_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
SACL_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
SACL_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
SACL_APPC_LINK	Char	16	477	492	Key to link together APPC records.
SACL_AUDIT_CODE	Char	11	494	504	Audit function code.
SACL_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
SACL_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
SACL_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
SACL_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
SACL_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
SACL_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
SACL_PATH_NAME	Char	1023	572	1594	The requested path name.
SACL_FILE_ID	Char	32	1596	1627	File ID.
SACL_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
SACL_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
SACL_FILEPOOL	Char	8	1651	1658	SFS filepool containing the BFS file.
SACL_FILESPACE	Char	8	1660	1667	SFS filespace containing the BFS file.
SACL_INODE	Integer	10	1669	1678	Inode (file serial number)

Table 157. Format of the SETFACL record extension (event code 75) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
SACL_SCID	Integer	10	1680	1689	File SCID
SACL_DCE_LINK	Char	16	1691	1706	Link to connect DCE records that originate from a DCE request
SACL_AUTH_TYPE	Char	13	1708	1720	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
SACL_DFLT_PROCESS	Yes/No	4	1722	1725	Default z/OS UNIX security environment in effect
SACL_UTK_NETW	Char	8	1727	1734	Port of entry network name
SACL_X500_SUBJECT	Char	255	1736	1990	Subject's name associated with this request.
SACL_X500_ISSUER	Char	255	1992	2246	Issuer's name associated with this request
SACL_ACL_TYPE	Char	8	2248	2255	What type of ACL is this? Valid values are "ACCESS", "FILEMOD", and "DIRMOD".
SACL_OPTYPE	Char	8	2257	2264	ACL entry operation. Valid values are "ADD", "MODIFY", and "DELETE".
SACL_ENTRY_TYPE	Char	3	2266	2268	ACL entry type. Valid values are "UID" and "GID".
SACL_ENTRY_ID	Integer	10	2270	2279	UID or GID value in the ACL entry.
SACL_OLD_READ	Yes/No	4	2281	2284	Was the READ bit on for this entry? (blank when SACL_OPTYPE is ADD)
SACL_OLD_WRITE	Yes/No	4	2286	2289	Was the WRITE bit on for this entry? (blank when SACL_OPTYPE is ADD)
SACL_OLD_EXECUTE	Yes/No	4	2291	2294	Was the EXECUTE bit on for this entry? (blank when SACL_OPTYPE is ADD)
SACL_NEW_READ	Yes/No	4	2296	2299	Was the READ bit on for this entry? (blank when SACL_OPTYPE is DELETE)
SACL_NEW_WRITE	Yes/No	4	2301	2304	Was the WRITE bit on for this entry? (blank when SACL_OPTYPE is DELETE)
SACL_NEW_EXECUTE	Yes/No	4	2306	2309	Was the EXECUTE bit on for this entry? (blank when SACL_OPTYPE is DELETE)
SACL_SECL	Char	8	2311	2318	Security label of the resource.
SACL_SERV_POENAME	Char	64	2320	2383	SERVAUTH resource or profile name.
SACL_CTX_USER	Char	510	2385	2894	Authenticated user name.
SACL_CTX_REG	Char	255	2896	3150	Authenticated user registry name.
SACL_CTX_HOST	Char	128	3152	3279	Authenticated user host name.
SACL_CTX_MECH	Char	16	3281	3296	Authenticated user authentication mechanism object identifier (OID).
SACL_IDID_USER	Char	985	3298	4282	Authenticated distributed user name.
SACL_IDID_REG	Char	1021	4284	5304	Authenticated distributed user registry name.

Table 158. Event qualifiers for SETFACL records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	ACL entry added, modified, or deleted.
INSAUTH	01	Caller does not have authority to change ACL of the specified file.
INSSECL	02	Insufficient security label.

The DELFACL record extension

Table 159 on page 321 describes the format of a record that is created by deleting an access control list of a z/OS UNIX file.

The event qualifiers that can be associated with an access control list deletion event are shown in Table 160 on page 322.

Table 159. Format of the DELFACL record extension (event code 76)					
Field name	Type	Length	Position		Comments
			Start	End	
DACL_CLASS	Char	8	282	289	Class name.
DACL_USER_NAME	Char	20	291	310	The name associated with the user ID.
DACL_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
DACL_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
DACL_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
DACL_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
DACL_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
DACL_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
DACL_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
DACL_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
DACL_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
DACL_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
DACL_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
DACL_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
DACL_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
DACL_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
DACL_UTK_SECL	Char	8	386	393	The security label of the user.
DACL_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
DACL_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
DACL_UTK_SNODE	Char	8	413	420	The submitting node.
DACL_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
DACL_UTK_SPOE	Char	8	431	438	The port of entry.
DACL_UTK_SPCCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DACL_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
DACL_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
DACL_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
DACL_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
DACL_APPC_LINK	Char	16	477	492	Key to link together APPC records.
DACL_AUDIT_CODE	Char	11	494	504	Audit function code.
DACL_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
DACL_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
DACL_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).

Table 159. Format of the DELFACL record extension (event code 76) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
DACL_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
DACL_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
DACL_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
DACL_PATH_NAME	Char	1023	572	1594	The requested path name.
DACL_FILE_ID	Char	32	1596	1627	File ID.
DACL_FILE_OWN_UID	Integer	10	1629	1638	The owner z/OS UNIX user identifier (UID) associated with the file.
DACL_FILE_OWN_GID	Integer	10	1640	1649	The owner z/OS UNIX group identifier (GID) associated with the file.
DACL_FILEPOOL	Char	8	1651	1658	SFS filepool containing the BFS file.
DACL_FILESPACE	Char	8	1660	1667	SFS filespace containing the BFS file.
DACL_INODE	Integer	10	1669	1678	Inode (file serial number)
DACL_SCID	Integer	10	1680	1689	File SCID
DACL_DCE_LINK	Char	16	1691	1706	Link to connect DCE records that originate from a DCE request
DACL_AUTH_TYPE	Char	13	1708	1720	Defines the type of request. Valid values are: "SERVER", "AUTH_CLIENT", and "UNAUTH_CLIENT".
DACL_DFLT_PROCESS	Yes/No	4	1722	1725	Default z/OS UNIX security environment in effect
DACL_UTK_NETW	Char	8	1727	1734	Port of entry network name
DACL_X500_SUBJECT	Char	255	1736	1990	Subject's name associated with this request.
DACL_X500_ISSUER	Char	255	1992	2246	Issuer's name associated with this request
DACL_ACL_TYPE	Char	8	2248	2255	What type of ACL is this? Valid values are "ACCESS", "FILEMOD", and "DIRMOD".
DACL_SECL	Char	8	2257	2264	Security label of the resource.
DACL_SERV_POENAME	Char	64	2266	2329	SERVAUTH resource or profile name.
DACL_CTX_USER	Char	510	2331	2840	Authenticated user name.
DACL_CTX_REG	Char	255	2842	3096	Authenticated user registry name.
DACL_CTX_HOST	Char	128	3098	3225	Authenticated user host name.
DACL_CTX_MECH	Char	16	3227	3242	Authenticated user authentication mechanism object identifier (OID).
DACL_IDID_USER	Char	985	3244	4228	Authenticated distributed user name.
DACL_IDID_REG	Char	1021	4230	5250	Authenticated distributed user registry name.

Table 160. Event qualifiers for DELFACL records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Entire ACL removed.
INSAUTH	01	Caller does not have authority to remove ACL of the specified file.
INSSECL	02	Insufficient security label.

The SETFSECL record extension

Table 161 on page 323 describes the format of a record that is created by setting the security label of a z/OS UNIX file.

The event qualifiers that can be associated with a set file security label event are shown in Table 162 on page 324.

Table 161. Format of the SETFSECL record extension (event code 77)					
Field name	Type	Length	Position		Comments
			Start	End	
SSCL_CLASS	Char	8	282	289	Class name.
SSCL_USER_NAME	Char	20	291	310	The name associated with the user ID.
SSCL_NEWSECL	Char	8	312	319	New security label.
SSCL_OLDSECL	Char	8	321	328	Old security label.
SSCL_UTK_ENCR	Yes/No	4	330	333	Is the UTOKEN associated with this user encrypted?
SSCL_UTK_PRE19	Yes/No	4	335	338	Is this a pre-1.9 token?
SSCL_UTK_VERPROF	Yes/No	4	340	343	Is the VERIFYX propagation flag set?
SSCL_UTK_NJEUNUSR	Yes/No	4	345	348	Is this the NJE undefined user?
SSCL_UTK_LOGUSR	Yes/No	4	350	353	Is UAUDIT specified for this user?
SSCL_UTK_SPECIAL	Yes/No	4	355	358	Is this a SPECIAL user?
SSCL_UTK_DEFAULT	Yes/No	4	360	363	Is this a default token?
SSCL_UTK_UNKNUSR	Yes/No	4	365	368	Is this an undefined user?
SSCL_UTK_ERROR	Yes/No	4	370	373	Is this user token in error?
SSCL_UTK_TRUSTED	Yes/No	4	375	378	Is this user a part of the trusted computed base (TCB)?
SSCL_UTK_SESSTYPE	Char	8	380	387	The session type of this session.
SSCL_UTK_SURROGAT	Yes/No	4	389	392	Is this a surrogate user?
SSCL_UTK_REMOTE	Yes/No	4	394	397	Is this a remote job?
SSCL_UTK_PRIV	Yes/No	4	399	402	Is this a privileged user ID?
SSCL_UTK_SECL	Char	8	404	411	The security label of the user.
SSCL_UTK_EXECNODE	Char	8	413	420	The execution node of the work.
SSCL_UTK_SUSER_ID	Char	8	422	429	The submitting user ID.
SSCL_UTK_SNODE	Char	8	431	438	The submitting node.
SSCL_UTK_SGRP_ID	Char	8	440	447	The submitting group name.
SSCL_UTK_SPOE	Char	8	449	456	The port of entry.
SSCL_UTK_SPCCLASS	Char	8	458	465	Class of the port of entry.
SSCL_UTK_USER_ID	Char	8	467	474	User ID associated with the record.
SSCL_UTK_GRP_ID	Char	8	476	483	Group name associated with the record.
SSCL_UTK_DFT_GRP	Yes/No	4	485	488	Is a default group assigned?
SSCL_UTK_DFT_SECL	Yes/No	4	490	493	Is a default security label assigned?
SSCL_AUDIT_CODE	Char	11	495	505	Audit function code.
SSCL_OLD_REAL_UID	Integer	10	507	516	Old real z/OS UNIX user identifier (UID).
SSCL_OLD_EFF_UID	Integer	10	518	527	Old effective z/OS UNIX user identifier (UID).
SSCL_OLD_SAVED_UID	Integer	10	529	538	Old saved z/OS UNIX user identifier (UID).

Table 161. Format of the SETFSECL record extension (event code 77) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
SSCL_OLD_REAL_GID	Integer	10	540	549	Old real z/OS UNIX group identifier (GID).
SSCL_OLD_EFF_GID	Integer	10	551	560	Old effective z/OS UNIX group identifier (GID).
SSCL_OLD_SAVED_GID	Integer	10	562	571	Old saved z/OS UNIX group identifier (GID).
SSCL_PATH_NAME	Char	1023	573	1595	The requested path name.
SSCL_FILE_ID	Char	32	1597	1628	File ID.
SSCL_FILE_OWN_UID	Integer	10	1630	1639	The owner z/OS UNIX user identifier (UID) associated with the file.
SSCL_FILE_OWN_GID	Integer	10	1641	1650	The owner z/OS UNIX group identifier (GID) associated with the file.
SSCL_DFLT_PROCESS	Yes/No	4	1652	1655	Default z/OS UNIX security environment in effect.
SSCL_UTK_NETW	Char	8	1657	1664	Port of entry network name.
SSCL_X500_SUBJECT	Char	255	1666	1920	Subject's name associated with this request.
SSCL_X500_ISSUER	Char	255	1922	2176	Issuer's name associated with this request.
SSCL_SERV_POENAME	Char	64	2178	2241	SERVAUTH resource or profile name.
SSCL_CTX_USER	Char	510	2243	2752	Authenticated user name.
SSCL_CTX_REG	Char	255	2754	3008	Authenticated user registry name.
SSCL_CTX_HOST	Char	128	3010	3137	Authenticated user host name.
SSCL_CTX_MECH	Char	16	3139	3154	Authenticated user authentication mechanism object identifier (OID).
SSCL_IDID_USER	Char	985	3156	4140	Authenticated distributed user name.
SSCL_IDID_REG	Char	1021	4142	5162	Authenticated distributed user registry name.

Table 162. Event qualifiers for SETFSECLrRecords		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Security label set successful.
NOTAUTH	01	Caller does not have authority to set security label.

The WRITEDWN record extension

Table 163 on page 324 describes the format of a record that is created by setting the write-down privilege.

The event qualifiers that can be associated with a set file write-down privilege event are shown in Table 164 on page 325.

Table 163. Format of the WRITEDWN record extension (event code 78)					
Field name	Type	Length	Position		Comments
			Start	End	
WDWN_USER_NAME	Char	20	282	301	The name associated with the user ID.
WDWN_UTK_ENCR	Yes/No	4	303	306	Is the UTKEN associated with this user encrypted?
WDWN_UTK_PRE19	Yes/No	4	308	311	Is this a pre-1.9 token?
WDWN_UTK_VERPROF	Yes/No	4	313	316	Is the VERIFYX propagation flag set?

Table 163. Format of the WRITEDWN record extension (event code 78) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
WDWN_UTK_NJEUNUSR	Yes/No	4	318	321	Is this the NJE undefined user?
WDWN_UTK_LOGUSR	Yes/No	4	323	326	Is UAUDIT specified for this user?
WDWN_UTK_SPECIAL	Yes/No	4	328	331	Is this a SPECIAL user?
WDWN_UTK_DEFAULT	Yes/No	4	333	336	Is this a default token?
WDWN_UTK_UNKNUSR	Yes/No	4	338	341	Is this an undefined user?
WDWN_UTK_ERROR	Yes/No	4	343	346	Is this user token in error?
WDWN_UTK_TRUSTED	Yes/No	4	348	351	Is this user a part of the trusted computed base (TCB)?
WDWN_UTK_SESSTYPE	Char	8	353	360	The session type of this session.
WDWN_UTK_SURROGAT	Yes/No	4	362	365	Is this a surrogate user?
WDWN_UTK_REMOTE	Yes/No	4	367	370	Is this a remote job?
WDWN_UTK_PRIV	Yes/No	4	372	375	Is this a privileged user ID?
WDWN_UTK_SECL	Char	8	377	384	The security label of the user.
WDWN_UTK_EXECNODE	Char	8	386	393	The execution node of the work.
WDWN_UTK_USUSER_ID	Char	8	395	402	The submitting user ID.
WDWN_UTK_SNODE	Char	8	404	411	The submitting node.
WDWN_UTK_SGRP_ID	Char	8	413	420	The submitting group name.
WDWN_UTK_SPOE	Char	8	422	429	The port of entry.
WDWN_UTK_SPCCLASS	Char	8	431	438	Class of the port of entry.
WDWN_UTK_USER_ID	Char	8	440	447	User ID associated with the record.
WDWN_UTK_GRP_ID	Char	8	449	456	Group name associated with the record.
WDWN_UTK_DFT_GRP	Yes/No	4	458	461	Is a default group assigned?
WDWN_UTK_DFT_SECL	Yes/No	4	463	466	Is a default security label assigned?
WDWN_UTK_NETW	Char	8	468	475	Port of entry network name.
WDWN_X500_SUBJECT	Char	255	477	731	Subject's name associated with this request.
WDWN_X500_ISSUER	Char	255	733	987	Issuer's name associated with this request.
WDWN_SERV_POENAME	Char	64	989	1052	SERVAUTH resource or profile name.
WDWN_CTX_USER	Char	510	1054	1563	Authenticated user name.
WDWN_CTX_REG	Char	255	1565	1819	Authenticated user registry name.
WDWN_CTX_HOST	Char	128	1821	1948	Authenticated user host name.
WDWN_CTX_MECH	Char	16	1950	1965	Authenticated user authentication mechanism object identifier (OID).
WDWN_IDID_USER	Char	985	1967	2951	Authenticated distributed user name.
WDWN_IDID_REG	Char	1021	2953	3973	Authenticated distributed user registry name.

Table 164. Event qualifiers for WRITEDWN records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	Success.
NOTAUTH	01	Caller does not have authority to set write-down privilege.

The PKIDPUBR record extension

Table 165 on page 326 describes the format of a record that is created by CRL publication.

The event qualifiers that can be associated with a CRL publication event are shown in Table 166 on page 326.

Table 165. Format of the PKIDPUBR record extension (event code 79)					
Field name	Type	Length	Position		Comments
			Start	End	
PKDP_CRL_SER_NUM	Char	255	282	536	CRL serial number
PKDP_ISSUERS_DN	Char	255	538	792	CRL issuer's distinguished name
PKDP_ISSUING_DP_DN	Char	255	794	1048	CRL's issuing distribution point distinguished name
PKDP_THIS_DATE	Date	10	1050	1059	CRL's date of issue
PKDP_THIS_TIME	Time	8	1061	1068	CRL's time of issue
PKDP_NEXT_DATE	Date	10	1070	1079	CRL's expiration date (issue date of next CRL)
PKDP_NEXT_TIME	Time	8	1081	1088	CRL's expiration time (issue time of next CRL)
PKDP_PUBLISH_DATE	Date	10	1090	1099	CRL's date of publish
PKDP_PUBLISH_TIME	Time	8	1101	1108	CRL's time of publish
PKDP_ISSUING_URI	Char	1024	1110	2133	CRL's issuing distribution point URI

Table 166. Event qualifiers for PKIDPUBR records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful publication of revocation information

The RPKIRESP record extension

Table 167 on page 326 describes the format of a record that is created by the PKI Services responder when a request for certificate status is made.

The event qualifiers that can be associated with RPKIRESP records are shown in Table 168 on page 327.

Table 167. Format of the RPKIRESP record extension (event code 80)					
Field name	Type	Length	Position		Comments
			Start	End	
RPKO_LOGSTRING	Char	255	282	536	Logstring parameter
RPKO_USER_NAME	Char	20	538	557	The name associated with the user ID
RPKO_UTK_ENCR	Yes/No	4	559	562	Is the UTKEN associated with this user encrypted?
RPKO_UTK_PRE19	Yes/No	4	564	567	Is this a pre-1.9 token?
RPKO_UTK_VERPROF	Yes/No	4	569	572	Is the VERIFYX propagation flag set?
RPKO_UTK_NJEUNUSR	Yes/No	4	574	577	Is this the NJE undefined user?
RPKO_UTK_LOGUSR	Yes/No	4	579	582	Is UAUDIT specified for this user?
RPKO_UTK_SPECIAL	Yes/No	4	584	587	Is this a SPECIAL user?
RPKO_UTK_DEFAULT	Yes/No	4	589	592	Is this a default token?
RPKO_UTK_UNKNUSR	Yes/No	4	594	597	Is this an undefined user?
RPKO_UTK_ERROR	Yes/No	4	599	602	Is this user token in error?

Table 167. Format of the RPKIRESP record extension (event code 80) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RPKO_UTK_TRUSTED	Yes/No	4	604	607	Is this user part of the trusted computer base (TCB)?
RPKO_UTK_SESSTYPE	Char	8	609	616	The session type of this session
RPKO_UTK_SURROGAT	Yes/No	4	618	621	Is this a surrogate user?
RPKO_UTK_REMOTE	Yes/No	4	623	626	Is this a remote job?
RPKO_UTK_PRIV	Yes/No	4	628	631	Is this a privileged user ID?
RPKO_UTK_SECL	Char	8	633	640	The security label of the user.
RPKO_UTK_EXECNODE	Char	8	642	649	The execution node of the work.
RPKO_UTK_SUSUSER_ID	Char	8	651	658	The submitting user ID.
RPKO_UTK_SNODE	Char	8	660	667	The submitting node.
RPKO_UTK_SGRP_ID	Char	8	669	676	The submitting group name.
RPKO_UTK_SPOE	Char	8	678	685	The port of entry.
RPKO_UTK_SPCLASS	Char	8	687	694	Class of the POE.
RPKO_UTK_USER_ID	Char	8	696	703	User ID associated with the record.
RPKO_UTK_GRP_ID	Char	8	705	712	Group name associates with the record.
RPKO_UTK_DFT_GROUP	Yes/No	4	714	717	Is a default group assigned?
RPKO_UTK_DFT_SECL	Yes/No	4	719	722	Is a default security label assigned?
RPKO_UTK_NETW	Char	8	724	731	The port of entry network name.
RPKO_X500_SUBJECT	Char	255	733	987	Subject's name associated with this event.
RPKO_X500_ISSUER	Char	255	989	1243	Issuer's name associated with this event.
RPKO_SERV_POENAME	Char	64	1245	1308	SERVAUTH resource or profile name.
RPKO_RESPONSE	Char	1024	1310	2333	Responses from OCSP.
RPKO_CA_DOMAIN	Char	8	2335	2342	Domain name of target PKI Services instance.
RPKO_CTX_USER	Char	510	2344	2853	Authenticated user name.
RPKO_CTX_REG	Char	255	2855	3109	Authenticated user registry name.
RPKO_CTX_HOST	Char	128	3111	3238	Authenticated user host name.
RPKO_CTX_MECH	Char	16	3240	3255	Authenticated user authentication mechanism object identifier (OID).
RPKO_IDID_USER	Char	985	3257	4241	Authenticated distributed user name.
RPKO_IDID_REG	Char	1021	4243	5263	Authenticated distributed user registry name.

Table 168. Event qualifiers for RPKIRESP records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful RESPOND request
INSAUTH	01	Insufficient authority for RESPOND

The PassTicket evaluation (PTEVAL) record extension

Table 169 on page 328 describes the format of a record that is created when a PassTicket is evaluated.

The event qualifiers that can be associated with PassTicket evaluation records are shown in [Table 170 on page 329](#).

<i>Table 169. Format of the PassTicket evaluation record extension (event code 81)</i>					
Field name	Type	Length	Position		Comments
			Start	End	
PTEV_APPLICATION	Char	8	282	289	Application name used in PassTicket operation.
PTEV_TARGET_USER	Char	8	291	298	User ID for which the PassTicket operation was performed. This is not the user who called the PassTicket service.
PTEV_USER_NAME	Char	20	300	319	The name associated with the caller of the PassTicket service. This is not the same user as PTEV_TARGET_USER.
PTEV_UTK_ENCR	Yes/No	4	321	324	Is the UTOKEN associated with this user encrypted?
PTEV__UTK_PRE19	Yes/No	4	326	329	Is this a pre-1.9 token?
PTEV_UTK_VERPROF	Yes/No	4	331	334	Is the VERIFYX propagation flag set?
PTEV_UTK_NJEUNUSR	Yes/No	4	336	339	Is this the NJE undefined user?
PTEV_UTK_LOGUSR	Yes/No	4	341	344	Is UAUDIT specified for this user?
PTEV_UTK_SPECIAL	Yes/No	4	346	349	Is this a SPECIAL user?
PTEV_UTK_DEFAULT	Yes/No	4	351	354	Is this a default token?
PTEV_UTK_UNKNUSR	Yes/No	4	356	359	Is this an undefined user?
PTEV_UTK_ERROR	Yes/No	4	361	364	Is this user token in error?
PTEV_UTK_TRUSTED	Yes/No	4	366	369	Is this user part of the trusted computing base (TCB)?
PTEV_UTK_SESTYPE	Char	8	371	378	The session type of this session.
PTEV_UTK_SURROGAT	Yes/No	4	380	383	Is this a surrogate user?
PTEV_UTK_REMOTE	Yes/No	4	385	388	Is this a remote job?
PTEV_UTK_PRIV	Yes/No	4	390	393	Is this a privileged user ID?
PTEV_UTK_SECL	Char	8	395	402	The security label of the user.
PTEV_UTK_EXECNODE	Char	8	404	411	The execution node of the work.
PTEV_UTK_SUSER_ID	Char	8	413	420	The submitting user ID.
PTEV_UTK_SNODE	Char	8	422	429	The submitting node.
PTEV_UTK_SGRP_ID	Char	8	431	438	The submitting group name.
PTEV_UTK_SPOE	Char	8	440	447	The port of entry.
PTEV_UTK_SPCLASS	Char	8	449	456	Class of the POE. Valid values are TERMINAL, CONSOLE, JESINPUT and APPCPORT.
PTEV_UTK_USER_ID	Char	8	458	465	User ID associated with the record.
PTEV_UTK_GRP_ID	Char	8	467	474	Group name associated with the record.
PTEV_UTK_DFT_GRP	Yes/No	4	476	479	Is a default group assigned?
PTEV_UTK_DFT_SECL	Yes/No	4	481	484	Is a default security label assigned?
PTEV_LPT_EVAL	Yes/No	4	486	489	The supplied password was evaluated as a legacy PassTicket.
PTEV_LPT_SUCC	Yes/No	4	491	494	The legacy PassTicket was evaluated successfully.
PTEV_EPT_UPPER_EVAL	Yes/No	4	496	499	The supplied Password was evaluated as an enhanced PassTicket type UPPER.
PTEV_EPT_UPPER_SUCC	Yes/No	4	501	504	The supplied Password was evaluated successfully as an enhanced PassTicket type UPPER.

Table 169. Format of the PassTicket evaluation record extension (event code 81) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
PTEV_EPT_MIXED_EVAL	Yes/No	4	506	509	The supplied Password was evaluated as an enhanced PassTicket type MIXED.
PTEV_EPT_MIXED_SUCC	Yes/No	4	511	514	The supplied Password was evaluated successfully as an enhanced PassTicket type MIXED.
PTEV_REPLAY_FAILURE	Yes/No	4	516	519	Failure due to replay attempt.
PTEV_RESERVED_08	Yes/No	4	521	524	Reserved for IBM's use.
PTEV_RESERVED_09	Yes/No	4	526	529	Reserved for IBM's use.
PTEV_RESERVED_10	Yes/No	4	531	534	Reserved for IBM's use.
PTEV_RESERVED_11	Yes/No	4	536	539	Reserved for IBM's use.
PTEV_RESERVED_12	Yes/No	4	541	544	Reserved for IBM's use.
PTEV_RESERVED_13	Yes/No	4	546	549	Reserved for IBM's use.
PTEV_RESERVED_14	Yes/No	4	551	443	Reserved for IBM's use.
PTEV_RESERVED_15	Yes/No	4	556	559	Reserved for IBM's use.
PTEV_RESERVED_16	Yes/No	4	561	564	Reserved for IBM's use.
PTEV_APPL_NAME	Char	8	566	573	Application name used to evaluate the PassTicket.
PTEV_EVAL_RSN1	Char	8	575	582	Evaluation Return Code. Expressed as hexadecimal number.
PTEV_EVAL_RSN2	Char	8	584	591	Evaluation Reason Code. Expressed as hexadecimal number.

Table 170. Event qualifiers for PTEVAL records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	PassTicket evaluation succeeded.
FAILURE	01	PassTicket evaluation failed.

The PassTicket generation (PTCREATE) record extension

Table 171 on page 329 describes the format of a record that is created when a PassTicket is generated.

The event qualifiers that can be associated with PassTicket generation records are shown in Table 172 on page 331.

Table 171. Format of the PassTicket generation record extension (event code 82)					
Field name	Type	Length	Position		Comments
			Start	End	
PTCR_APPLICATION	Char	8	282	289	Application name used in PassTicket operation.
PTCR_TARGET_USER	Char	8	291	298	User ID for which the PassTicket operation was performed. This is not the user who called the PassTicket service.
PTCR_USER_NAME	Char	20	300	319	The name associated with the caller of the PassTicket service. This is not the same user as PTCR_TARGET_USER
PTCR_UTK_ENCR	Yes/No	4	321	324	Is the UTKEN associated with this user encrypted?
PTCR__UTK_PRE19	Yes/No	4	326	329	Is this a pre-1.9 token?

Table 171. Format of the PassTicket generation record extension (event code 82) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
PTCR_UTK_VERPROF	Yes/No	4	331	334	Is the VERIFYX propagation flag set?
PTCR_UTK_NJEUNUSR	Yes/No	4	336	339	Is this the NJE undefined user?
PTCR_UTK_LOGUSR	Yes/No	4	341	344	Is UAUDIT specified for this user?
PTCR_UTK_SPECIAL	Yes/No	4	346	349	Is this a SPECIAL user?
PTCR_UTK_DEFAULT	Yes/No	4	351	354	Is this a default token?
PTCR_UTK_UNKNUSR	Yes/No	4	356	359	Is this an undefined user?
PTCR_UTK_ERROR	Yes/No	4	361	364	Is this user token in error?
PTCR_UTK_TRUSTED	Yes/No	4	366	369	Is this user part of the trusted computing base (TCB)?
PTCR_UTK_SESTYPE	Char	8	371	378	The session type of this session.
PTCR_UTK_SURROGAT	Yes/No	4	380	383	Is this a surrogate user?
PTCR_UTK_REMOTE	Yes/No	4	385	388	Is this a remote job?
PTCR_UTK_PRIV	Yes/No	4	390	393	Is this a privileged user ID?
PTCR_UTK_SECL	Char	8	395	402	The security label of the user.
PTCR_UTK_EXECNODE	Char	8	404	411	The execution node of the work.
PTCR_UTK_SUSER_ID	Char	8	413	420	The submitting user ID.
PTCR_UTK_SNODE	Char	8	422	429	The submitting node.
PTCR_UTK_SGRP_ID	Char	8	431	438	The submitting group name.
PTCR_UTK_SPOE	Char	8	440	447	The port of entry.
PTCR_UTK_SPCLASS	Char	8	449	456	Class of the POE. Valid values are TERMINAL, CONSOLE, JESINPUT and APPCPORT.
PTCR_UTK_USER_ID	Char	8	458	465	User ID associated with the record.
PTCR_UTK_GRP_ID	Char	8	467	474	Group name associated with the record.
PTCR_UTK_DFT_GRP	Yes/No	4	476	479	Is a default group assigned?
PTCR_UTK_DFT_SECL	Yes/No	4	481	484	Is a default security label assigned?
PTCR_LPT	Yes/No	4	486	489	Generate of a legacy PassTicket was attempted.
PTCR_RESERVED_02	Yes/No	4	491	494	Reserved for IBM's use.
PTCR_EPT_UPPER	Yes/No	4	496	499	Generate of an enhanced PassTicket type UPPER was attempted.
PTCR_RESERVED_04	Yes/No	4	501	504	Reserved for IBM's use.
PTCR_EPT_MIXED	Yes/No	4	506	509	Generate of an enhanced PassTicket type MIXED was attempted.
PTCR_RESERVED_06	Yes/No	4	511	514	Reserved for IBM's use.
PTCR_RESERVED_07	Yes/No	4	516	519	Reserved for IBM's use.
PTCR_RESERVED_08	Yes/No	4	521	524	Reserved for IBM's use.
PTCR_RESERVED_09	Yes/No	4	526	529	Reserved for IBM's use.
PTCR_RESERVED_10	Yes/No	4	531	534	Reserved for IBM's use.
PTCR_RESERVED_11	Yes/No	4	536	539	Reserved for IBM's use.
PTCR_RESERVED_12	Yes/No	4	541	544	Reserved for IBM's use.
PTCR_RESERVED_13	Yes/No	4	546	549	Reserved for IBM's use.
PTCR_RESERVED_14	Yes/No	4	551	554	Reserved for IBM's use.

Table 171. Format of the PassTicket generation record extension (event code 82) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
PTCR_RESERVED_15	Yes/No	4	556	559	Reserved for IBM's use.
PTCR_RESERVED_16	Yes/No	4	561	564	Reserved for IBM's use.
PTCR_APPL_NAME	Char	8	566	573	Application Name used to generate the PassTicket.
PTCR_GEN_RSN1	Char	8	575	582	Generation Return Code. Expressed as hexadecimal number.
PTCR_GEN_RSN2	Char	8	584	591	Generation Reason Code. Expressed as hexadecimal number.

Table 172. Event qualifiers for PTCREATE records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	PassTicket was generated.
FAILURE	01	PassTicket generation failed.

The RPKISCEP record extension

Table 173. Format of the RPKISCEP record extension (event code 83)					
Field name	Type	Length	Position		Comments
			Start	End	
RPKS_LOGSTRING	Char	255	282	536	Logstring parameter.
RPKS_USER_NAME	Char	20	538	557	The name associated with the user ID.
RPKS_UTK_ENCR	Yes/No	4	559	562	Is the UTOKEN associated with this user encrypted?
RPKS_UTK_PRE19	Yes/No	4	564	567	Is this a pre-1.9 token?
RPKS_UTK_VERPROF	Yes/No	4	569	572	Is the VERIFYX propagation flag set?
RPKS_UTK_NJEUNUSR	Yes/No	4	574	577	Is this the NJE undefined user?
RPKS_UTK_LOGUSR	Yes/No	4	579	582	Is UAUDIT specified for this user?
RPKS_UTK_SPECIAL	Yes/No	4	584	587	Is this a SPECIAL user?
RPKS_UTK_DEFAULT	Yes/No	4	589	592	Is this a default token?
RPKS_UTK_UNKNUSR	Yes/No	4	594	597	Is this an undefined user?
RPKS_UTK_ERROR	Yes/No	4	599	602	Is this user token in error?
RPKS_UTK_TRUSTED	Yes/No	4	604	607	Is this user a part of the TCB?
RPKS_UTK_SESTYPE	Char	8	609	616	The session type of this session.
RPKS_UTK_SURROGAT	Yes/No	4	618	621	Is this a surrogate user?
RPKS_UTK_REMOTE	Yes/No	4	623	626	Is this a remote job?
RPKS_UTK_PRIV	Yes/No	4	628	631	Is this a privileged user ID?
RPKS_UTK_SECL	Char	8	633	640	The security label of the user.
RPKS_UTK_EXECNODE	Char	8	642	649	The execution node of the work.
RPKS_UTK_SUSER_ID	Char	8	651	658	The submitting user ID.
RPKS_UTK_SNODE	Char	8	660	667	The submitting node
RPKS_UTK_SGRP_ID	Char	8	669	676	The submitting group name.

Table 173. Format of the RPKISCEP record extension (event code 83) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RPKS_UTK_SPOE	Char	8	678	685	The port of entry.
RPKS_UTK_SPCCLASS	Char	8	687	694	Class of the POE.
RPKS_UTK_USER_ID	Char	8	696	703	User ID associated with the record.
RPKS_UTK_GRP_ID	Char	8	705	712	Group name associated with the record.
RPKS_UTK_DFT_GRP	Yes/No	4	714	717	Is a default group assigned?
RPKS_UTK_DFT_SECL	Yes/No	4	719	722	Is a default security label assigned?
RPKS_SERIAL_NUMBER	Char	255	724	978	Certificate serial number.
RPKS_ISSUERS_DN	Char	255	980	1234	Certificate issuer's distinguished name.
RPKS_UTK_NETW	Char	8	1236	1243	The port of entry network name.
RPKS_X500_SUBJECT	Char	255	1245	1499	Subject's name associated with this event.
RPKS_X500_ISSUER	Char	255	1501	1755	Issuer's name associated with this event.
RPKS_KEYUSAGE	Char	64	1757	1820	Requested certificate KeyUsage.
RPKS_NOTBEFOR_DATE	Char	10	1822	1831	Requested certificate NotBefore date.
RPKS_NOTAFTER_DATE	Char	10	1833	1842	Requested certificate NotAfter date.
RPKS_SUBJECTS_DN	Char	255	1844	2098	Certificate subject's distinguished name.
RPKS_ALT_IP	Char	64	2100	2163	Requested ALTNAME IP address.
RPKS_ALT_URI	Char	255	2165	2419	Requested ALTNAME URI.
RPKS_ALT_EMAIL	Char	100	2421	2520	Requested ALTNAME email.
RPKS_ALT_DOMAIN	Char	100	2522	2621	Requested ALTNAME Domain.
RPKS_CERT_ID	Char	56	2623	2678	IRRSPX00 Certificate ID.
RPKS_HOSTID_MAP	Char	1024	2680	3703	Reserved for IBM's use.
RPKS_REQUESTOR	Char	32	3705	3736	Requester's name - SCEP transaction ID.
RPKS_PASS_PHRASE	Yes/No	4	3738	3741	Requester specified a pass phrase.
RPKS_NOTIFY_EMAIL	Char	64	3743	3806	Reserved for IBM's use.
RPKS_EXTKEYUSAGE	Char	255	3808	4062	Requested Extended KeyUsage.
RPKS_SERV_POENAME	Char	64	4064	4127	SERVAUTH resource or profile name.
RPKS_ALT_OTHER	Char	1024	4129	5152	Requested ALTNAME OtherName.
RPKS_CA_DOMAIN	Char	8	5154	5161	Domain name of target PKI Services instance.
RPKS_CTX_USER	Char	510	5163	5672	Authenticated user name.
RPKS_CTX_REG	Char	255	5674	5928	Authenticated user registry name.
RPKS_CTX_HOST	Char	128	5930	6057	Authenticated user host name.
RPKS_CTX_MECH	Char	16	6059	6074	Authenticated user authentication mechanism object identifier (OID).
RPKS_IDID_USER_UTF8	Char	246	6076	6321	Authenticated distributed user name in UTF-8.
RPKS_IDID_USER_EBCDIC	Char	738	6323	7060	Authenticated distributed user name in EBCDIC.
RPKS_IDID_REG_UTF8	Char	255	7062	7316	Authenticated distributed registry name in UTF-8.
RPKS_IDID_REG_EBCDIC	Char	765	7318	8082	Authenticated distributed registry name in EBCDIC.
RPKS_CUSTOM_EXT	Char	1024	8084	9107	Customized extension.
RPKS_RECORD_LINK	Char	32	9109	9140	Field to link audit records together.

Table 173. Format of the RPKISCEP record extension (event code 83) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RPKS_CERT_FGRPRNT	Char	64	9142	9205	Subject Certificate SHA256 fingerprint in printable hex value

Table 174. Event qualifiers for RPKISCEP records		
Event qualifier	Event qualifier number	Event description
SUCCAUTO	00	Successful AutoApprove PKCSReq request.
SUCCADIM	01	Successful AdminApprove PKCSReq request
SUCCGETI	02	Successful GetCertInitial request
REJECTED	03	Rejected PKCSReq or GetCertInitial request
INCORRECT	04	Incorrect SCEP transaction ID specified for GetCertInitial
INSAUTH	05	Insufficient authority for SCEPREQ

The RDATAUPD record extension

Table 175. Format of the RDATAUPD record extension (event code 84)					
Field name	Type	Length	Position		Comments
			Start	End	
RPUT_USER_NAME	Char	20	282	301	The name associated with the user ID.
RPUT_UTK_ENCR	Yes/No	4	303	306	Is the UTKEN associated with this user encrypted?
RPUT_UTK_PRE19	Yes/No	4	308	311	Is this a pre-1.9 token?
RPUT_UTK_VERPROF	Yes/No	4	313	316	Is the VERIFYX propagation flag set?
RPUT_UTK_NJEUNUSR	Yes/No	4	318	321	Is this the NJE undefined user?
RPUT_UTK_LOGUSR	Yes/No	4	323	326	Is UAUDIT specified for this user?
RPUT_UTK_SPECIAL	Yes/No	4	328	331	Is this a SPECIAL user?
RPUT_UTK_DEFAULT	Yes/No	4	333	336	Is this a default token?
RPUT_UTK_UNKNUSR	Yes/No	4	338	341	Is this an undefined user?
RPUT_UTK_ERROR	Yes/No	4	343	346	Is this user token in error?
RPUT_UTK_TRUSTED	Yes/No	4	348	351	Is this user a part of the TCB?
RPUT_UTK_SESSTYPE	Char	8	353	360	The session type of this session.
RPUT_UTK_SURROGAT	Yes/No	4	362	365	Is this a surrogate user?
RPUT_UTK_REMOTE	Yes/No	4	367	370	Is this a remote job?
RPUT_UTK_PRIV	Yes/No	4	372	375	Is this a privileged user ID?
RPUT_UTK_SECL	Char	8	377	384	The security label of the user.
RPUT_UTK_EXECNODE	Char	8	386	393	The execution node of the work.
RPUT_UTK_SUSER_ID	Char	8	395	402	The submitting user ID.
RPUT_UTK_SNODE	Char	8	404	411	The submitting node.
RPUT_UTK_SGRP_ID	Char	8	413	420	The submitting group name.

Format of unloaded SMF type 80 data

Table 175. Format of the RDATAUPD record extension (event code 84) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RPUT_UTK_SPOE	Char	8	422	429	The port of entry.
RPUT_UTK_SPCLASS	Char	8	431	438	Class of the POE.
RPUT_UTK_USER_ID	Char	8	440	447	User ID associated with the record.
RPUT_UTK_GRP_ID	Char	8	449	456	Group name associated with the record.
RPUT_UTK_DFT_GRP	Yes/No	4	458	461	Is a default group assigned?
RPUT_UTK_DFT_SECL	Yes/No	4	463	466	Is a default security label assigned?
RPUT_SERIAL_NUMBER	Char	255	468	722	Certificate serial number.
RPUT_ISSUERS_DN	Char	255	724	978	Certificate issuer's distinguished name.
RPUT_RING_NAME	Char	237	980	1216	Ring name.
RPUT_UTK_NEW	Char	8	1218	1225	The port of entry network name.
RPUT_X500_SUBJECT	Char	255	1227	1481	Subject's name associated with this event.
RPUT_X500_ISSUER	Char	255	1483	1737	Issuer's name associated with this event.
RPUT_CERT_OWNER	Char	8	1739	1746	Certificate owner.
RPUT_CERT_LABEL	Char	32	1748	1779	Certificate label.
RPUT_SUBJECTS_DN	Char	255	1781	2035	Certificate subject's distinguished name.
RPUT_RING_OWNER	Char	8	2037	2044	Ring owner.
RPUT_ATTR_REUSE	Yes/No	4	2046	2049	Is the reuse attribute on?
RPUT_ATTR_TRUST	Yes/No	4	2051	2054	Is the trust attribute on?
RPUT_ATTR_HITRUST	Yes/No	4	2056	2059	Is the hightrust attribute on?
RPUT_ATTR_DELETE	Yes/No	4	2061	2064	Is the delete attribute on?
RPUT_CERT_USAGE	Char	8	2066	2073	Certificate usage in ring.
RPUT_CERT_DEFAULT	Yes/No	4	2075	2078	Is it the default certificate?
RPUT_PRIVATE_KEY	Yes/No	4	2080	2083	Is there a private key?
RPUT_ATTR_NOTRUST	Yes/No	4	2085	2088	Is the notrust attribute on?
RPUT_ATTR_DELFROMRING	Yes/No	4	2090	2093	Is the attribute that determines whether a certificate is to be deleted, even if it is connected to the rings, turned on?
RPUT_ATTR_DELFORCE	Yes/No	4	2095	2098	Is the delete force attribute on?
RPUT_SOURCE_LABEL	Char	32	2100	2131	Source certificate label
RPUT_CERT_FGRPRNT	Char	64	2133	2196	Certificate SHA256 fingerprint

Table 176. Event qualifiers for RDATAUPD records

Event qualifier	Event qualifier number	Event description
SUCCNEW	00	Successful NewRing.
INAUNEW	01	Not authorized to call NewRing.

Table 176. Event qualifiers for RDATAUPD records (continued)

Event qualifier	Event qualifier number	Event description
SUCCPUT	02	Successful DataPut.
INAUPUT	03	Not authorized to call DataPut.
SUCCRMV	04	Successful DataRemove.
INAURMV	05	Not authorized to call DataRemove
SUCCDEL	06	Successful DelRing.
INAUDEL	07	Not authorized to call DelRing.
SUCCALT	08	Successful DataAlter.
INAUALT	09	Not authorized to call DataAlter.

The PKIAURNW record extension

Table 177. Format of the PKIAURNW record extension (event code 85)

Field name	Type	Length	Position		Comments
			Start	End	
PKRN_SERIAL_NUMBER	Char	255	282	536	Certificate serial number.
PKRN_ISSUERS_DN	Char	255	538	792	Certificate issuer's distinguished name.
PKRN_NOTBEFOR_DATE	Char	10	794	803	Requested certificate's NotBefore date.
PKRN_NOTAFTER_DATE	Char	10	805	814	Requested certificate's NotAfter date.
PKRN_SUBJECTS_DN	Char	255	816	1070	Certificate subject's distinguished name.
PKRN_REQUESTOR	Char	32	1072	1103	Requester's name.
PKRN_PREV_SERIAL	Char	255	1105	1359	Previous serial number of the certificate.
PKRN_NOTIFY_EMAIL	Char	64	1361	1424	Email address for notification purposes.
PKRN_CA_DOMAIN	Char	8	1426	1433	Domain name of target PKI Services instance.
PKRN_EXIT_PATH	Char	256	1435	1690	Full path name of the exit.
PKRN_CERT_FGRPRNT	Char	64	1692	1755	Subject Certificate SHA256 fingerprint in printable hex value
PKRN_ISU_CERT_FGRPRNT	Char	64	1757	1820	Issuer Certificate SHA256 fingerprint in hex value
PKRN_PREV_CERT_FGRPRNT	Char	64	1822	1885	Previous Certificate SHA256 fingerprint in printable hex value

Table 178. Event qualifiers for PKIAURNW records

Event qualifier	Event qualifier number	Event description
SUCCRNEW	00	Successful Renew.

The PGMVERIFY record extension

Table 179 on page 336 describes the format of a record that is created by the R_PgmSignVer callable service.

The event qualifiers that can be associated with an R_PgmSignVer callable service are shown in Table 180 on page 337.

Table 179. Format of the PGMVERIFY record extension (event code 86)					
Field name	Type	Length	Position		Comments
			Start	End	
PGMV_RES_NAME	Char	255	282	536	Name of program being verified.
PGMV_VOL	Char	6	538	543	Volume containing the program.
PGMV_LOGSTRING	Char	255	545	799	Logstring parameter.
PGMV_USER_NAME	Char	20	801	820	The name associated with the user ID.
PGMV_UTK_ENCR	Yes/No	4	822	825	Is the UTOKEN associated with this user encrypted?
PGMV_UTK_PRE19	Yes/No	4	827	830	Is this a pre-1.9 token?
PGMV_UTK_VERPROF	Yes/No	4	832	835	Is the VERIFYX propagation flag set?
PGMV_UTK_NJEUNUSR	Yes/No	4	837	840	Is this the NJE undefined user?
PGMV_UTK_LOGUSR	Yes/No	4	842	845	Is UAUDIT specified for this user?
PGMV_UTK_SPECIAL	Yes/No	4	847	850	Is this a SPECIAL user?
PGMV_UTK_DEFAULT	Yes/No	4	852	855	Is this a default token?
PGMV_UTK_UNKNUSR	Yes/No	4	857	860	Is this an undefined user?
PGMV_UTK_ERROR	Yes/No	4	862	865	Is this user token in error?
PGMV_UTK_TRUSTED	Yes/No	4	867	870	Is this user a part of the TCB?
PGMV_UTK_SESSTYPE	Char	8	872	879	The session type of this session.
PGMV_UTK_SURROGAT	Yes/No	4	881	884	Is this a surrogate user?
PGMV_UTK_REMOTE	Yes/No	4	886	889	Is this a remote job?
PGMV_UTK_PRIV	Yes/No	4	891	894	Is this a privileged user ID?
PGMV_UTK_SECL	Char	8	896	903	The security label of the user.
PGMV_UTK_EXECNODE	Char	8	905	912	The execution node of the work.
PGMV_UTK_SUSER_ID	Char	8	914	921	The submitting user ID.
PGMV_UTK_SNODE	Char	8	923	930	The submitting node.
PGMV_UTK_SGRP_ID	Char	8	932	939	The submitting group name.
PGMV_UTK_SPOE	Char	8	941	948	The port of entry.
PGMV_UTK_SPCLASS	Char	8	950	957	Class of the POE.
PGMV_UTK_USER_ID	Char	8	959	966	User ID associated with the record.
PGMV_UTK_GRP_ID	Char	8	968	975	Group name associated with the record.
PGMV_UTK_DFT_GRP	Yes/No	4	977	980	Is a default group assigned?
PGMV_UTK_DFT_SECL	Yes/No	4	982	985	Is a default security label assigned?
PGMV_PDS_DSN	Char	44	987	1030	Partitioned data set name containing the program.
PGMV_UTK_NETW	Char	8	1032	1039	The port of entry network name.
PGMV_X500_SUBJECT	Char	255	1041	1295	Subject's name associated with this event.
PGMV_X500_ISSUER	Char	255	1297	1551	Issuer's name associated with this event.

Table 179. Format of the PGMVERIFY record extension (event code 86) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
PGMV_SERV_POENAME	Char	64	1553	1616	SERVAUTH resource or profile name.
PGMV_CTX_USER	Char	510	1618	2127	Authenticated user name.
PGMV_CTX_REG	Char	255	2129	2383	Authenticated user registry name.
PGMV_CTX_HOST	Char	128	2385	2512	Authenticated user host name.
PGMV_CTX_MECH	Char	16	2514	2529	Authenticated user authentication mechanism object identifier (OID).
PGMV_ROOT_DN	Char	255	2531	2785	Root signing certificate subject's distinguished name.
PGMV_SIGNER_DN	Char	255	2787	3041	Program signing certificate subject's distinguished name.
PGMV_MOD_LOADED	Yes/No	4	3043	3046	Module loaded?
PGMV_SIGN_TIME	Time	8	3048	3055	Time module was signed.
PGMV_SIGN_DATE	Date	10	3057	3066	Date module was signed.
PGMV_EXPIR_DATE	Date	10	3068	3077	Date at which module signature certificate chain expires.
PGMV_IDID_USER	Char	985	3079	4063	Authenticated distributed user name.
PGMV_IDID_REG	Char	1021	4065	5085	Authenticated distributed user registry name.

Table 180. Event qualifiers for PGMVERIFY records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful signature verification.
NOTRUST	01	Signature appears valid but root CA certificate not trusted.
INVALSIG	02	Module signature failed verification.
INCORCHN	03	Module certificate chain incorrect.
NOTSIGND	04	Signature required but module not signed.
SIGREMOV	05	Signature required but signature has been removed.
VERNOTLD	06	Program verification module not loaded. Program verification was not available when attempt was made to load this program.
SLFTSTFL	07	Algorithmic self test failed.

The RACMAP record extension

Table 181. Format of the RACMAP record extension (event code 87)					
Field name	Type	Length	Position		Comments
			Start	End	
RACM_USER_NAME	Char	20	282	301	The name associated with the user ID.
RACM_UTK_ENCR	Yes/No	4	303	306	Is the UTKEN associated with this user encrypted?
RACM_UTK_PRE19	Yes/No	4	308	311	Is this a pre-1.9 token?
RACM_UTK_VERPROF	Yes/No	4	313	316	Is the VERIFYX propagation flag set?
RACM_UTK_NJEUNUSR	Yes/No	4	318	321	Is this the NJE undefined user?

Table 181. Format of the RACMAP record extension (event code 87) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RACM_UTK_LOGUSR	Yes/No	4	323	326	Is UAUDIT specified for this user?
RACM_UTK_SPECIAL	Yes/No	4	328	331	Is this a SPECIAL user?
RACM_UTK_DEFAULT	Yes/No	4	333	336	Is this a default token?
RACM_UTK_UNKNUSR	Yes/No	4	338	341	Is this an undefined user?
RACM_UTK_ERROR	Yes/No	4	343	346	Is this user token in error?
RACM_UTK_TRUSTED	Yes/No	4	348	351	Is this user a part of the trusted computing base?
RACM_UTK_SESSTYPE	Char	8	353	360	The session type of this session.
RACM_UTK_SURROGAT	Yes/No	4	362	365	Is this a surrogate user?
RACM_UTK_REMOTE	Yes/No	4	367	370	Is this a remote job?
RACM_UTK_PRIV	Yes/No	4	372	375	Is this a privileged user ID?
RACM_UTK_SECL	Char	8	377	384	The security label of the user.
RACM_UTK_EXECNODE	Char	8	386	393	The execution node of the work.
RACM_UTK_SUSER_ID	Char	8	395	402	The submitting user ID.
RACM_UTK_SNODE	Char	8	404	411	The submitting node.
RACM_UTK_SGRP_ID	Char	8	413	420	The submitting group name.
RACM_UTK_SPOE	Char	8	422	429	The port of entry.
RACM_UTK_SPCLASS	Char	8	431	438	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
RACM_UTK_USER_ID	Char	8	440	447	User ID associated with the record.
RACM_UTK_GRP_ID	Char	8	449	456	Group name associated with the record.
RACM_UTK_DFT_GRP	Yes/No	4	458	461	Is a default group assigned?
RACM_UTK_DFT_SECL	Yes/No	4	463	466	Is a default security label assigned?
RACM_SPECIFIED	Char	1024	468	1491	The keywords specified on the RACMAP command.
RACM_UTK_NETW	Char	8	1493	1500	The port of entry network name.
RACM_X500_SUBJECT	Char	255	1502	1756	Subject's name associated with this event.
RACM_X500_ISSUER	Char	255	1758	2012	Issuer's name associated with this event.
RACM_SERV_POENAME	Char	64	2014	2077	SERVAUTH resource or profile name.
RACM_CTX_USER	Char	510	2079	2588	Authenticated user name.
RACM_CTX_REG	Char	255	2590	2844	Authenticated user registry name.
RACM_CTX_HOST	Char	128	2846	2973	Authenticated user host name.
RACM_CTX_MECH	Char	16	2975	2990	Authenticated user authentication mechanism object identifier (OID).
RACM_IDID_USER	Char	985	2992	3976	Authenticated distributed user name.
RACM_IDID_REG	Char	1021	3978	4998	Authenticated distributed user registry name.

Table 182. Event qualifiers for RACMAP records

Event qualifier	Event qualifier number	Event description
SUCCESS	00	Success.
NOTAUTH	01	Caller does not have authority.

The AUTOPROF record extension

Table 183. Format of the AUTOPROF record extension (event code 88)

Field name	Type	Length	Position		Comments
			Start	End	
AUTO_CLASS	Char	8	282	289	Class name.
AUTO_USER_NAME	Char	20	291	310	The name associated with the user ID.
AUTO_UTK_ENCR	Yes/No	4	312	315	Is the UTKEN associated with this user encrypted?
AUTO_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
AUTO_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
AUTO_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
AUTO_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
AUTO_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
AUTO_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
AUTO_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
AUTO_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
AUTO_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
AUTO_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
AUTO_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
AUTO_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
AUTO_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
AUTO_UTK_SECL	Char	8	386	393	The security label of the user.
AUTO_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
AUTO_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
AUTO_UTK_SNODE	Char	8	413	420	The submitting node.
AUTO_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
AUTO_UTK_SPOE	Char	8	431	438	The port of entry.
AUTO_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
AUTO_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
AUTO_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
AUTO_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
AUTO_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?

Table 183. Format of the AUTOPROF record extension (event code 88) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
AUTO_APPC_LINK	Char	16	477	492	A key to link together audit record together for a user's APPC transaction processing work.
AUTO_AUDIT_CODE	Char	11	494	504	Audit function code.
AUTO_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
AUTO_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
AUTO_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
AUTO_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
AUTO_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
AUTO_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
AUTO_DFLT_PROCESS	Yes/No	4	572	575	Default z/OS UNIX security environment in effect.
AUTO_UTK_NETW	Char	8	577	584	The port of entry network name.
AUTO_X500_SUBJECT	Char	255	586	840	Subject's name associated with this event.
AUTO_X500_ISSUER	Char	255	842	1096	Issuer's name associated with this event.
AUTO_SERV_POENAME	Char	64	1098	1161	SERVAUTH resource or profile name
AUTO_CTX_USER	Char	510	1163	1672	Authenticated user name.
AUTO_CTX_REG	Char	255	1674	1928	Authenticated user registry name.
AUTO_CTX_HOST	Char	128	1930	2057	Authenticated user host name.
AUTO_CTX_MECH	Char	16	2059	2074	Authenticated user authentication mechanism object identifier (OID).
AUTO_MOD_SERVICE	Char	20	2076	2095	Service or process name.
AUTO_MOD_CLASS	Char	8	2097	2104	Class for automatically updated profile.
AUTO_MOD_PROF	Char	255	2106	2360	Auto-updated profile name.
AUTO_MOD_DATA	Char	4000	2362	6361	Auto-updated profile data.
AUTO_IDID_USER	Char	985	6363	7347	Authenticated distributed user name.
AUTO_IDID_REG	Char	1021	7349	8369	Authenticated distributed user registry name.

Table 184. Event qualifiers for AUTOPROF records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful profile update.

The RPKIQREC record extension

Table 185. Format of the RPKIQREC record extension (event code 89)					
Field name	Type	Length	Position		Comments
			Start	End	
RPKQ_APPL	Char	8	282	289	The application data or application name from the original request.
RPKQ_LOGSTRING	Char	255	291	545	Logstring parameter.
RPKQ_USER_NAME	Char	20	547	566	The name associated with the user ID.
RPKQ_UTK_ENCR	Yes/No	4	568	571	Is the UTKEN associated with this user encrypted?
RPKQ_UTK_PRE19	Yes/No	4	573	576	Is this a pre-1.9 token?
RPKQ_UTK_VERPROF	Yes/No	4	578	581	Is the VERIFYX propagation.
RPKQ_UTK_NJEUNUSR	Yes/No	4	583	586	Is this the NJE undefined user?
RPKQ_UTK_LOGUSR	Yes/No	4	588	591	Is UAUDIT specified for this user?
RPKQ_UTK_SPECIAL	Yes/No	4	593	596	Is this a SPECIAL user?
RPKQ_UTK_DEFAULT	Yes/No	4	598	601	Is this a default token?
RPKQ_UTK_UNKNUSR	Yes/No	4	603	606	Is this an undefined user?
RPKQ_UTK_ERROR	Yes/No	4	608	611	Is this user token in error?
RPKQ_UTK_TRUSTED	Yes/No	4	613	616	Is this user a part of the TCB?
RPKQ_UTK_SESSTYPE	Char	8	618	625	The session type of this session.
RPKQ_UTK_SURROGAT	Yes/No	4	627	630	Is this a surrogate user?
RPKQ_UTK_REMOTE	Yes/No	4	632	635	Is this a remote job?
RPKQ_UTK_PRIV	Yes/No	4	637	640	Is this a privileged user ID?
RPKQ_UTK_SECL	Char	8	642	649	The SECLABEL of the user.
RPKQ_UTK_EXECNODE	Char	8	651	658	The execution node of the work.
RPKQ_UTK_SUSER_ID	Char	8	660	667	The submitting user ID.
RPKQ_UTK_SNODE	Char	8	669	676	The submitting node.
RPKQ_UTK_SGRP_ID	Char	8	678	685	The submitting group name.
RPKQ_UTK_SPOE	Char	8	687	694	The port of entry.
RPKQ_UTK_SPCLASS	Char	8	696	703	Class of the POE.
RPKQ_UTK_USER_ID	Char	8	705	712	User ID associated with the record.
RPKQ_UTK_GRP_ID	Char	8	714	721	Group name associated with the record.
RPKQ_UTK_DFT_GRP	Yes/No	4	723	726	Is a default group assigned?
RPKQ_UTK_DFT_SECL	Yes/No	4	728	731	Is a default SECLABEL assigned?
RPKQ_SERIAL_NUMBER	Char	255	733	987	Certificate serial number.
RPKQ_ISSUERS_DN	Char	255	989	1243	Certificate issuer's distinguished name.
RPKQ_UTK_NETW	Char	8	1245	1252	The port of entry network name.
RPKQ_X500_SUBJECT	Char	255	1254	1508	Subject's name associated with this event.
RPKQ_X500_ISSUER	Char	255	1510	1764	Issuer's name associated with this event.

Table 185. Format of the RPKIQREC record extension (event code 89) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RPKQ_NOTBEFOR_DATE	Char	10	1766	1775	Requested certificate NotBefore date.
RPKQ_NOTAFTER_DATE	Char	10	1777	1786	Requested certificate NotAfter date.
RPKQ_SUBJECTS_DN	Char	255	1788	2042	Certificate subject's distinguished name.
RPKQ_REQUESTOR	Char	32	2044	2075	Requester's email address.
RPKQ_SERV_POENAME	Char	64	2077	2140	SERVAUTH resource or profile name.
RPKQ_CA_DOMAIN	Char	8	2142	2149	Domain name of the target PKI Services instance.
RPKQ_CTX_USER	Char	510	2151	2660	Authenticated user name.
RPKQ_CTX_REG	Char	255	2662	2916	Authenticated user registry name.
RPKQ_CTX_HOST	Char	128	2918	3045	Authenticated user host name.
RPKQ_CTX_MECH	Char	16	3047	3062	Authenticated user authentication mechanism object identifier (OID).
RPKQ_KEY_ID	Char	40	3064	3103	Hash of the public key generated by PKI Services.
RPKQ_IDID_USER	Char	985	3105	4089	Authenticated distributed user name.
RPKQ_IDID_REG	Char	1021	4091	5111	Authenticated distributed user registry name.

Table 186. Event qualifiers for RPKIQREC records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful user QRECOVER request.
INSAUTH	01	Insufficient authority for user QRECOVER.

The PKIGENC record extension

Table 187 on page 342 describes the format of a record that is created by PKIGENC.

The event qualifiers that can be associated with PKIGENC records are shown in Table 188 on page 343.

Table 187. Format of the PKIGENC record extension (event code 90)					
Field name	Type	Length	Position		Comments
			Start	End	
PKGC_CERT_FGRPRNT	Char	64	282	345	Subject Certificate SHA256 fingerprint in printable hex value
PKGC_ISU_CERT_FGRPRNT	Char	64	347	410	Issuer Certificate SHA256 fingerprint in printable hex value
PKGC_SERIAL_NUM	Char	255	412	666	Certificate serial number
PKGC_ISSUERS_DN	Char	255	668	922	Certificate issuer distinguished name
PKGC_SUBJECTS_DN	Char	255	924	1178	Certificate subject distinguished name
PKGC_NOTBEFOR_DATE	Char	10	1180	1189	Certificate start date
PKGC_NOTAFTER_DATE	Char	10	1191	1200	Certificate expiration date

Table 187. Format of the PKIGENC record extension (event code 90) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
PKGC_CA_DOMAIN	Char	8	1202	1209	Domain name of target PKI Services instance

Table 188. Event qualifiers for PKIGENC records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Successful certificate GENCERT request.

The PRLIMIT record extension

Table 189 on page 343 describes the format of a record that is created by the prlimit() API.

The event qualifiers that can be associated with prlimit() are shown in Table 190 on page 344.

Table 189. Format of the prlimit record extension (event code 91)					
Field name	Type	Length	Position		Comments
			Start	End	
PRLM_CLASS	Char	8	282	289	Class name.
PRLM_USER_NAME	Char	20	291	310	The name associated with the user ID.
PRLM_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
PRLM_UTK_PRE19	Yes/No	4	317	320	Is this a pre-1.9 token?
PRLM_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
PRLM_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
PRLM_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
PRLM_UTK_SPECIAL	Yes/No	4	337	340	Is this a SPECIAL user?
PRLM_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
PRLM_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
PRLM_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
PRLM_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
PRLM_UTK_SESSTYPE	Yes/No	8	362	369	The session type of this session.
PRLM_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
PRLM_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
PRLM_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
PRLM_UTK_SECL	Char	8	386	393	The security label of the user.
PRLM_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
PRLM_UTK_SUSER_IDL	Char	8	404	411	The submitting user ID.
PRLM_UTK_SNODE	Char	8	413	420	The submitting node.
PRLM_UTK_SGRP_ID	Char	8	422	429	The submitting group name.
PRLM_UTK_SPOE	Char	8	431	438	The port of entry.
PRLM_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
PRLM_UTK_USER_ID	Char	8	449	456	User ID associated with the record.

Table 189. Format of the prlimit record extension (event code 91) (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
PRLM_UTK_GRP_ID	Char	8	458	465	Group name associated with the record.
PRLM_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
PRLM_UTK_DFT_SECL	Yes/No	4	472	475	Is a default security label assigned?
PRLM_APPC_LINK	Char	16	477	492	Key to link together APPC records.
PRLM_AUDIT_CODE	Char	11	494	504	Audit function code.
PRLM_OLD_REAL_UID	Integer	10	506	515	Old real z/OS UNIX user identifier (UID).
PRLM_OLD_EFF_UID	Integer	10	517	526	Old effective z/OS UNIX user identifier (UID).
PRLM_OLD_SAVED_UID	Integer	10	528	537	Old saved z/OS UNIX user identifier (UID).
PRLM_OLD_REAL_GID	Integer	10	539	548	Old real z/OS UNIX group identifier (GID).
PRLM_OLD_EFF_GID	Integer	10	550	559	Old effective z/OS UNIX group identifier (GID).
PRLM_OLD_SAVED_GID	Integer	10	561	570	Old saved z/OS UNIX group identifier (GID).
PRLM_TGT_REAL_UID	Integer	10	572	581	Target real z/OS UNIX user identifier (UID).
PRLM_TGT_EFF_UID	Integer	10	583	592	Target effective z/OS UNIX user identifier (UID).
PRLM_TGT_SAV_UID	Integer	10	594	603	Target saved z/OS UNIX user identifier (UID).
PRLM_TGT_REAL_GID	Integer	10	605	614	Target real z/OS UNIX group identifier (GID).
PRLM_TGT_EFF_GID	Integer	10	616	625	Target effective z/OS UNIX group identifier (GID).
PRLM_TGT_SAV_GID	Integer	10	627	636	Target saved z/OS UNIX group identifier (GID).
PRLM_TGT_PID	Integer	10	638	647	Target process ID.
PRLM_DFLT_PROCESS	Yes/No	4	649	652	Default z/OS UNIX security environment in effect.
PRLM_UTK_NETW	Char	8	654	661	The port of entry network name.
PRLM_X500_SUBJECT	Char	225	663	917	Subject's name associated with this event.
PRLM_X500_ISSUER	Char	255	919	1173	Issuer's name associated with this event.
PRLM_SECL	Char	8	1175	1182	Security label of the resource.
PRLM_SERV_POENAME	Char	64	1184	1247	SERVAUTH resource or profile name.
PRLM_CTX_USER	Char	510	1249	1758	Authenticated user name.
PRLM_CTX_REG	Char	255	1760	2014	Authenticated user registry name.
PRLM_CTX_HOST	Char	128	2016	2143	Authenticated user host name.
PRLM_CTX_MECH	Char	16	2145	2160	Authenticated user authentication mechanism object identifier (OID).
PRLM_IDID_USER	Char	985	2162	3146	Authenticated distributed user name.
PRLM_IDID_REG	Char	1021	3148	4168	Authenticated distributed user registry name.

Table 190. Event qualifiers for PRLIMIT records		
Event qualifier	Event qualifier number	Event description
SUCCESS	00	Prlimit successful.
NOTAUTH	01	Not authorized to issue prlimit.
INSSECL	02	Insufficient security label.

Security event record extension (unloaded)

Table 191 on page 345 describes the format of a record that is created by the security event.

The event qualifiers that can be associated with security event records are shown in Table 192 on page 345.

Table 191. Format of the security event record extension (event code 92)					
Field name	Type	Length	Position		Comments
			Start	End	
SECE_USERID	Char	8	282	289	User ID associated with the event.
SECE_MODULE	Char	8	291	298	Module (calling routine) that was associated with the event when the failure occurred.

Table 192. Event qualifiers for security event records		
Event qualifier	Event qualifier number	Event description
CONTAIND	00	User ID containment.

Chapter 7. The format of the unloaded SMF type 81 data

RACF writes a type 81 record at the completion of the initialization of RACF. [Table 193 on page 347](#) describes the format of the unloaded version of this record.

Table 193. Format of the unloaded SMF type 81 records					
Field name	Type	Length	Position		Comments
			Start	End	
RINI_EVENT_TYPE	Char	8	1	8	The type of the event. Set to "RACFINIT".
RINI_RESERVED_01	Char	8	10	17	This field is reserved and is set to blanks to allow a common alignment with other unloaded SMF records.
RINI_TIME_WRITTEN	Time	8	19	26	Time that the record was written to SMF.
RINI_DATE_WRITTEN	Date	10	28	37	Date that the record was written to SMF.
RINI_SYSTEM_SMFID	Char	4	39	42	SMF system ID of the system from which the record originates.
RINI_DATASET_NAME	Char	44	44	87	Name of the RACF database for this IPL.
RINI_DATASET_VOL	Char	6	89	94	Volume upon which the RACF data set resides
RINI_DATASET_UNIT	Char	3	96	98	Unit name of the RACF database.
RINI_UADS_NAME	Char	44	100	143	Name of the user attribute data set for this IPL.
RINI_UADS_VOL	Char	6	145	150	Volume upon which the user attribute data set resides.
RINI_RACINIT_STATS	Yes/No	4	152	155	Are RACINIT statistics recorded?
RINI_DATASET_STATS	Yes/No	4	157	160	Are data set statistics recorded?
RINI_RACINIT_PRE	Yes/No	4	162	165	Is there a RACROUTE REQUEST=VERIFY preprocessing exit (ICHRIX01)?
RINI_RACHECK_PRE	Yes/No	4	167	170	Is there a RACROUTE REQUEST=AUTH preprocessing exit (ICHRCX01)?
RINI_RACDEF_PRE	Yes/No	4	172	175	Is there a RACROUTE REQUEST=DEFINE preprocessing exit (ICHRDX01)?
RINI_RACINIT_POST	Yes/No	4	177	180	Is there a RACROUTE REQUEST=VERIFY postprocessing exit (ICHRIX02)?
RINI_RACHECK_POST	Yes/No	4	182	185	Is there a RACROUTE REQUEST=AUTH postprocessing exit (ICHRCX02)?
RINI_NEW_PWD_EXIT	Yes/No	4	187	190	Is there a new-password exit routine (ICHPWX01)?
RINI_TAPEVOL_STATS	Yes/No	4	192	195	Are tape volume statistics recorded?
RINI_DASD_STATS	Yes/No	4	197	200	Are DASD statistics recorded?
RINI_TERM_STATS	Yes/No	4	202	205	Are terminal statistics recorded?
RINI_CMD_EXIT	Yes/No	4	207	210	Is the command exit routine ICHCNX00 active? ICHCNX00 is invoked for RACF commands, the IRRUT100 utility, and by IRRXT00 when a RACROUTE REQUEST=EXTRACT is issued for CLASS=DATASET.
RINI_DEL_CMD_EXIT	Yes/No	4	212	215	Is the command exit routine ICHCCX00 active? ICHCCX00 is invoked for the DELGROUP, DELUSER, and REMOVE commands.
RINI_ADSP	Yes/No	4	217	220	Is ADSP active?

Table 193. Format of the unloaded SMF type 81 records (continued)

Field name	Type	Length	Position		Comments
			Start	End	
RINI_ENCRYPT_EXIT	Yes/No	4	222	225	Is the encryption exit ICHDEX01 active?
RINI_NAMING_CONV	Yes/No	4	227	230	Is the naming convention table ICHNCV00 present?
RINI_TAPEVOL	Yes/No	4	232	235	Is tape volume protection in effect?
RINI_DUP_DSNS	Yes/No	4	237	240	Are duplicate data set names allowed to be defined?
RINI_DASD	Yes/No	4	242	245	Is DASD volume protection in effect?
RINI_FRACHECK_PRE	Yes/No	4	247	250	Is the RACROUTE REQUEST=FASTAUTH preprocessing exit (ICHRFX01) active?
RINI_RACLIST_PRE	Yes/No	4	252	255	Is the RACROUTE REQUEST=LIST pre/postprocessing exit (ICHLX01) active?
RINI_RACLIST_SEL	Yes/No	4	257	260	Is the RACROUTE REQUEST=LIST selection exit (ICHLX02) active?
RINI_RACDEF_POST	Yes/No	4	262	265	Is the RACROUTE REQUEST=DEFINE postprocessing exit (ICHRDX02) active?
RINI_AUDIT_USER	Yes/No	4	267	270	Are user class profile changes being audited?
RINI_AUDIT_GROUP	Yes/No	4	272	275	Are group class profile changes being audited?
RINI_AUDIT_DATASET	Yes/No	4	277	280	Are data set class profile changes being audited?
RINI_AUDIT_TAPEVOL	Yes/No	4	282	285	Are tape volume class profile changes being audited?
RINI_AUDIT_DASDVOL	Yes/No	4	287	290	Are DASD volume class profile changes being audited?
RINI_AUDIT_TERM	Yes/No	4	292	295	Are terminal class profile changes being audited?
RINI_AUDIT_CMDVIOL	Yes/No	4	297	300	Are command violations being audited?
RINI_AUDIT_SPECIAL	Yes/No	4	302	305	Are special users being audited?
RINI_AUDIT_OPER	Yes/No	4	307	310	Are operations users being audited?
RINI_AUDIT_LEVEL	Yes/No	4	312	315	Is auditing by security level in effect?
RINI_ACEE_COMPRESS	Yes/No	4	317	320	Is the IRRACX01 exit in effect?
RINI_FASTAUTH_PRE	Yes/No	4	322	325	Is the FASTAUTH AR mode preprocessing exit (ICHRFX03) in effect?
RINI_FASTAUTH_POST	Yes/No	4	327	330	Is the FASTAUTH AR mode postprocessing exit (ICHRFX04) in effect?
RINI_TERM	Yes/No	4	332	335	Is terminal authorization checking in effect?
RINI_TERM_NONE	Yes/No	4	337	340	Are undefined terminals treated as UACC=NONE?
RINI_REALDSN	Yes/No	4	342	345	Is REALDSN in effect?
RINI_XBMALLRACF	Yes/No	4	347	350	Is the JES XBMALLRACF option in effect?
RINI_EARLYVERIFY	Yes/No	4	352	355	Is the JES EARLYVERIFY option in effect?
RINI_BATCHALLRACF	Yes/No	4	357	360	Is the JES BATCHALLRACF option in effect?
RINI_FRACHECK_POST	Yes/No	4	362	365	Is the RACROUTE REQUEST=FASTAUTH post processing exit (ICHRFX02) in effect?
RINI_PWD_INT	Integer	3	367	369	The maximum password interval.
RINI_SINGLE_DSN	Char	8	371	378	The single level data set name.
RINI_TAPEDSN	Yes/No	4	380	383	Is TAPEDSN in effect?
RINI_PROTECTALL	Yes/No	4	385	388	Is PROTECTALL in effect?
RINI_PROTECTALL_W	Yes/No	4	390	393	Is PROTECTALL warning in effect?

Table 193. Format of the unloaded SMF type 81 records (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RINI_ERASE	Yes/No	4	395	398	Is ERASE-ON-SCRATCH in effect?
RINI_ERASE_LEVEL	Yes/No	4	400	403	Is ERASE-ON-SCRATCH based on security level in effect?
RINI_ERASE_ALL	Yes/No	4	405	408	Is ERASE-ON-SCRATCH for all data sets in effect?
RINI_EGN	Yes/No	4	410	413	Is enhanced generic naming in effect?
RINI_WHEN_PROGRAM	Yes/No	4	415	418	Is access control by program in effect?
RINI_RETENTION	Integer	5	420	424	System retention period.
RINI_LEVEL_ERASE	Integer	5	426	430	Security level for ERASE-ON-SCRATCH.
RINI_LEVEL_AUDIT	Integer	5	432	436	Security level for auditing.
RINI_SECL_CTRL	Yes/No	4	438	441	Is SECLABELCONTROL in effect?
RINI_CATDSNS	Yes/No	4	443	446	Is CATDSNS in effect?
RINI_MLQUIET	Yes/No	4	448	451	Is MLQUIET in effect?
RINI_MLSTABLE	Yes/No	4	453	456	Is MLSTABLE in effect?
RINI_MLS	Yes/No	4	458	461	Is SETROPTS MLS (no write-down) in effect?
RINI_MLACTIVE	Yes/No	4	463	466	Is MLACTIVE in effect?
RINI_GENERIC_OWNER	Yes/No	4	468	471	Is GENERICOWNER in effect?
RINI_SECL_AUDIT	Yes/No	4	473	476	Is SECLABELAUDIT in effect?
RINI_SESSION_INT	Integer	5	478	482	Partner LU-verification session key interval.
RINI_NJE_NAME_ID	Char	8	484	491	JES NJE name user ID.
RINI_NJE_UDFND_ID	Char	8	493	500	JES UNDEFINEDUSER user ID.
RINI_COMPATMODE	Yes/No	4	502	505	Is COMPATMODE in effect?
RINI_CATDSNS_FAIL	Yes/No	4	507	510	Is CATDSNS failures in effect?
RINI_MLS_FAIL	Yes/No	4	512	515	Is MLS failures in effect?
RINI_MLACTIVE_FAIL	Yes/No	4	517	520	Is MLACTIVE failures in effect?
RINI_APPLAUD	Yes/No	4	522	525	Is APPLAUDIT in effect?
RINI_DFT_PRI	Char	3	527	529	Default primary language for the installation.
RINI_DFT_SEC	Char	3	531	533	Default secondary language for the installation.
RINI_RESERVED_02	Char	4	535	538	Reserved for IBM's use
RINI_ALL_CMD_EXIT	Yes/No	4	540	543	Did the exit for all commands (IRREVX01) have any active exit routines at IPL time?
RINI_ADDCREATOR	Yes/No	4	545	548	Is the SETROPTS ADDCREATOR option in effect?
RINI_ACEE_COMP_XM	Yes/No	4	550	553	Is the IRRACX02 exit in effect?
RINI_ENCRYPT_EXIT2	Yes/No	4	555	558	Is IRRDEX11 exit in effect?
RINI_PWD_HIST	Integer	3	560	562	The password history value.
RINI_PWD_REVOKE	Integer	3	564	566	The number of incorrect logon passwords before users are revoked.
RINI_PWD_WARN	Integer	3	568	570	The number of days before password expiry during which users receive a warning message.
RINI_PWDRULE1_MIN	Integer	1	572	572	Password syntax rule 1 minimum length.
RINI_PWDRULE1_MAX	Integer	1	574	574	Password syntax rule 1 maximum length.

Table 193. Format of the unloaded SMF type 81 records (continued)

Field name	Type	Length	Position		Comments
			Start	End	
RINI_PWDRULE1	Char	8	576	583	Password syntax rule 1.
RINI_PWDRULE2_MIN	Integer	1	585	585	Password syntax rule 2 minimum length.
RINI_PWDRULE2_MAX	Integer	1	587	587	Password syntax rule 2 maximum length.
RINI_PWDRULE2	Char	8	589	596	Password syntax rule 2.
RINI_PWDRULE3_MIN	Integer	1	598	598	Password syntax rule 3 minimum length.
RINI_PWDRULE3_MAX	Integer	1	600	600	Password syntax rule 3 maximum length.
RINI_PWDRULE3	Char	8	602	609	Password syntax rule 3.
RINI_PWDRULE4_MIN	Integer	1	611	611	Password syntax rule 4 minimum length.
RINI_PWDRULE4_MAX	Integer	1	613	613	Password syntax rule 4 maximum length.
RINI_PWDRULE4	Char	8	615	622	Password syntax rule 4.
RINI_PWDRULE5_MIN	Integer	1	624	624	Password syntax rule 5 minimum length.
RINI_PWDRULE5_MAX	Integer	1	626	626	Password syntax rule 5 maximum length.
RINI_PWDRULE5	Char	8	628	635	Password syntax rule 5.
RINI_PWDRULE6_MIN	Integer	1	637	637	Password syntax rule 6 minimum length.
RINI_PWDRULE6_MAX	Integer	1	639	639	Password syntax rule 6 maximum length.
RINI_PWDRULE6	Char	8	641	648	Password syntax rule 6.
RINI_PWDRULE7_MIN	Integer	1	650	650	Password syntax rule 7 minimum length.
RINI_PWDRULE7_MAX	Integer	1	652	652	Password syntax rule 7 maximum length.
RINI_PWDRULE7	Char	8	654	661	Password syntax rule 7.
RINI_PWDRULE8_MIN	Integer	1	663	663	Password syntax rule 8 minimum length.
RINI_PWDRULE8_MAX	Integer	1	665	665	Password syntax rule 8 maximum length.
RINI_PWDRULE8	Char	8	667	674	Password syntax rule 8.
RINI_INACTIVE	Integer	3	676	678	The number of days of inactivity before users are revoked.
RINI_GRPLIST	Yes/No	4	680	683	Is list-of-groups processing in effect?
RINI_MODEL_GDG	Yes/No	4	685	688	Is MODEL(GDG) in effect?
RINI_MODEL_USER	Yes/No	4	690	693	Is MODEL(USER) in effect?
RINI_MODEL_GROUP	Yes/No	4	695	698	Is MODEL(GROUP) in effect?
RINI_RSWI_INST_PWD	Yes/No	4	700	703	"Yes" if an installation-defined RVAR SWITCH password is in effect. "No" if the default RVAR SWITCH password is in effect.
RINI_RSTA_INST_PWD	Yes/No	4	705	708	"Yes" if an installation-defined RVAR STATUS password is in effect. "No" if the default RVAR STATUS password is in effect.
RINI_KERBLVL	Integer	3	710	712	Level of KERB segment processing in effect.
RINI_MLFS	Char	8	714	721	What is the status of the MLFSOBJ SETROPTS option? (Active or inactive.)
RINI_MLIPC	Char	8	723	730	What is the status of the SETROPTS MLIPCOBJ option? (Active or inactive)
RINI_MLNAMES	Yes/No	4	732	735	Is MLNAMES in effect?
RINI_SLBYSYS	Yes/No	4	737	740	Is SECLBYSYSTEM in effect?

Table 193. Format of the unloaded SMF type 81 records (continued)					
Field name	Type	Length	Position		Comments
			Start	End	
RINI_PWD_MIN	Integer	3	742	744	The password minimum change interval
RINI_PWD_MIXED	Yes/No	4	746	749	Are mixed case passwords allowed?
RINI_NEW_PHR_EXIT	Yes/No	4	751	754	Is the ICHPWX11 exit in effect?
RINI_FLD_VAL_EXIT	Yes/No	4	756	759	Did the field validation exit for custom fields (IRRVAF01) have any active exit routines at IPL time?
RINI_PWD_SPECIAL	Yes/No	4	761	764	Are special characters allowed in passwords?
RINI_PWD_ALG	Char	10	766	777	Algorithm that is used to encrypt passwords and password phrases. Possible values are "KDFAES" and "LEGACY".
RINI_ENHANCED_GENOWNER	Yes/No	4	779	782	Is ENHANCEDGENERICOWNER in effect?
RINI_PHR_INT	Integer	5	784	788	The password phrase interval.

The format of the unloaded SMF type 81 class data

Table 194 on page 351 describes the format of the class information that is contained in the SMF type 81 record.

Table 194. Format of the unloaded SMF type 81 class records					
Field Name	Type	Length	Position		Comments
			Start	End	
RINC_EVENT_TYPE	Char	8	1	8	The type of the event. Set to "CLASNAME".
RINC_RESERVED_01	Char	8	10	17	This field is reserved and is set to blanks to allow a common alignment with other unloaded SMF records.
RINC_TIME_WRITTEN	Time	8	19	26	Time that the record was written to SMF.
RINC_DATE_WRITTEN	Date	10	28	37	Date that the record was written to SMF.
RINC_SYSTEM_SMFID	Char	4	39	42	SMF system ID of the system from which the record originates.
RINC_CLASS_NAME	Char	8	44	51	The name of the class.
RINC_STATS	Yes/No	4	53	56	Are statistics collected for this class?
RINC_AUDIT	Yes/No	4	58	61	Is this class being audited?
RINC_ACTIVE	Yes/No	4	63	66	Is this class active?
RINC_GENERIC	Yes/No	4	68	71	Can generic profiles be defined in this class?
RINC_GENCMD	Yes/No	4	73	76	Is generic command processing enabled for this class?
RINC_GLOBAL	Yes/No	4	78	81	Is this class enabled for global access checking?
RINC_RACLIST	Yes/No	4	83	86	Has SETR RACLIST been issued for this class?
RINC_GENLIST	Yes/No	4	88	91	Has SETR GENLIST been issued for this class?
RINC_LOG_OPTIONS	Char	8	93	100	The LOGOPTIONS for the class. Valid values are "ALWAYS", "NEVER", "SUCCESS", "FAILURES", and "DEFAULT".

For more information on XML grammar and IRRADU00 record format, see:

- [“IRRADU00 record format” on page 169](#)
- [“XML grammar” on page 170](#)

- [“Steps for converting RACF field names to XML tag names” on page 170](#)

Chapter 8. The format of the unloaded SMF type 83 data

RACF writes a type 83 subtype 1 record for each data set that is affected by the change of a security label. Table 195 on page 353 describes the format of the unloaded version of this record.

Table 195. Format of the unloaded SMF type 83 records					
Field name	Type	Length	Position		Comments
			Start	End	
DSAF_EVENT_TYPE	Char	8	1	8	The type of the event. Set to "DSAF".
DSAF_RESERVED_01	Char	8	10	17	This field is reserved and is set to blanks to allow a common alignment with other unloaded SMF records.
DSAF_TIME_WRITTEN	Time	8	19	26	Time that the record was written to SMF.
DSAF_DATE_WRITTEN	Date	10	28	37	Date that the record was written to SMF.
DSAF_SYSTEM_SMFID	Char	4	39	42	SMF system ID of the system from which the record originates.
DSAF_SECL_LINK	Char	16	44	59	Key to link together the data sets affected by a change of security label and the command that caused the security label change.
DSAF_VIOLATION	Yes/No	4	61	64	Does this record represent a violation?
DSAF_USER_NDFND	Yes/No	4	66	69	Was this user not defined to RACF?
DSAF_USER_WARNING	Yes/No	4	71	74	Was this record created because of WARNING?
DSAF_EVT_USER_ID	Char	8	76	83	User ID associated with the event.
DSAF_EVT_GRP_ID	Char	8	85	92	Group name associated with the event.
DSAF_AUTH_NORMAL	Yes/No	4	94	97	Was normal authority checking a reason for access being allowed?
DSAF_AUTH_SPECIAL	Yes/No	4	99	102	Was special authority checking a reason for access being allowed?
DSAF_AUTH_OPER	Yes/No	4	104	107	Was operations authority checking a reason for access being allowed?
DSAF_AUTH_AUDIT	Yes/No	4	109	112	Was auditor authority checking a reason for access being allowed?
DSAF_AUTH_EXIT	Yes/No	4	114	117	Was exit checking a reason for access being allowed?
DSAF_AUTH_FAILSFT	Yes/No	4	119	122	Was failsoft checking a reason for access being allowed?
DSAF_AUTH_BYPASS	Yes/No	4	124	127	Was the use of the user ID *BYPASS* a reason for access being allowed?
DSAF_AUTH_TRUSTED	Yes/No	4	129	132	Was trusted authority checking a reason for access being allowed?
DSAF_LOG_CLASS	Yes/No	4	134	137	Was SETR AUDIT(class) checking a reason for this event to be recorded?
DSAF_LOG_USER	Yes/No	4	139	142	Was auditing requested for this user?
DSAF_LOG_SPECIAL	Yes/No	4	144	147	Was auditing requested for access granted due to the SPECIAL privilege?
DSAF_LOG_ACCESS	Yes/No	4	149	152	Did the profile indicate audit, or did FAILSOFT processing allow access, or did the RACHECK exit indicate auditing?

Table 195. Format of the unloaded SMF type 83 records (continued)

Field name	Type	Length	Position		Comments
			Start	End	
DSAF_LOG_RACINIT	Yes/No	4	154	157	Did the RACINIT fail?
DSAF_LOG_ALWAYS	Yes/No	4	159	162	Is this command always audited?
DSAF_LOG_CMDVIOL	Yes/No	4	164	167	Was this event audited due to CMDVIOL?
DSAF_LOG_GLOBAL	Yes/No	4	169	172	Was this event audited due to GLOBALAUDIT?
DSAF_TERM_LEVEL	Integer	3	174	176	The terminal level associated with this audit record.
DSAF_BACKOUT_FAIL	Yes/No	4	178	181	Did RACF fail in backing out the data?
DSAF_PROF_SAME	Yes/No	4	183	186	Was the profile the same at the end of this event?
DSAF_TERM	Char	8	188	195	The terminal associated with the event.
DSAF_JOB_NAME	Char	8	197	204	The job name associated with the event.
DSAF_READ_TIME	Time	8	206	213	The time that the job entered the system.
DSAF_READ_DATE	Date	10	215	224	The date that the job entered the system.
DSAF_SMF_USER_ID	Char	8	226	233	User ID from SMF common area. This value is managed by SMF and the SMF processing exits.
DSAF_LOG_LEVEL	Yes/No	4	235	238	Was this event audited due to SECLEVEL auditing?
DSAF_LOG_LOGOPT	Yes/No	4	240	243	Was this event audited due to SETR LOGOPTIONS auditing?
DSAF_LOG_SECL	Yes/No	4	245	248	Was this event audited due to SETR SECLABELAUDIT auditing?
DSAF_LOG_COMPATM	Yes/No	4	250	253	Was this event audited due to SETR COMPATMODE auditing?
DSAF_LOG_APPLAUD	Yes/No	4	255	258	Was this event audited due to SETR APPLAUDIT?
DSAF_USR_SECL	Char	8	260	267	The security label associated with this user.
DSAF_DATA_SET	Char	44	269	312	The name of the data set affected by the security label change.
DSAF_RESERVED_02	Char	2	314	315	Reserved for IBM's use.
DSAF_PROD_ID	Char	8	317	324	Short name for the product or component logging the event

Note: The format of unloaded SMF records, subtype 2 and above, is described in product-specific documentation.

For more information on XML grammar and IRRADU00 record format, see:

- [“IRRADU00 record format” on page 169](#)
- [“XML grammar” on page 170](#)
- [“Steps for converting RACF field names to XML tag names” on page 170](#)

Chapter 9. RACF database unload utility (IRRDBU00) records

For a description of the RACF database unload utility and instructions on how to run it, see [z/OS Security Server RACF Security Administrator's Guide](#).

Running the unload: When you run the database unload utility against a database that is active on a system that is a member of the RACF sysplex data sharing group, run the utility from a system in the group. Otherwise, the utility might produce unpredictable results.

IRRDBU00 record types

The database unload utility gives every record it creates a record type. This record type is a 4-byte identification number in the first four positions of every record. For details about the format of the identification number, see [“Format of the record type identification number”](#) on page 359.

The record types and their associated names are:

Record Type

Record Name

0100

Group Basic Data

0101

Group Subgroups

0102

Group Members

0103

Group Installation Data

0110

Group DFP Data

0120

Group OMVS Data

0130

Group OVM Data

0140

Reserved

0141

Group TME Data

0150

Reserved

0151

Group CSDATA Custom Fields

0200

User Basic Data

0201

User Categories

0202

User Classes

0203

User Group Connections

0204

User Installation Data

0205

User Connect Data

0206

User RRSF Data

0207

User Certificate Name

0208

User Associated Mappings

0209

User Associated Distributed Mappings

020A

User MFA Factor Data Record

020B

User MFA Policies Record

0210

User DFP Data

0220

User TSO Data

0230

User CICS Data

0231

User CICS Operator Classes

0232

User CICS RSL Keys

0233

User CICS TSL Keys

0240

User Language Data

0250

User OPERPARM Data

0251

User OPERPARM Scope

0260

User WORKATTR Data

0270

User OMVS Data

0280

User NETVIEW Segment

0281

User OPCLASS

0282

User DOMAINS

0290

User DCE Data

02A0

User OVM Data

02B0

User LNOTES Data

02C0	User NDS Data
02D0	User KERB Data
02E0	User PROXY Data
02F0	User EIM Data
02G0	Reserved
02G1	User CSDATA Custom Fields
0400	Data Set Basic Data
0401	Data Set Categories
0402	Data Set Conditional Access
0403	Data Set Volumes
0404	Data Set Access
0405	Data Set Installation Data
0410	Data Set DFP Data
0420	Reserved
0421	Data Set TME Data
0431	Data Set CSDATA Record
0500	General Resource Basic Data
0501	General Resource Tape Volume Data
0502	General Resource Categories
0503	General Resource Members
0504	General Resource Volumes
0505	General Resource Access
0506	General Resource Installation Data
0507	General Resource Conditional Access
0508	Filter Data

0509

General Resource Distributed Identity Mapping Data

0510

General Resource Session Data

0511

General Resource Session Entities

0520

General Resource DLF Data

0521

General Resource DLF Job Names

0530

General Resource SSIGNON Data Record

0540

General Resource Started Task Data

0550

General Resource SystemView Data

0560

General Resource Certificate Data

0561

General Resource Certificate References

0562

General Resource Key Ring Data

0570

General Resource TME Data

0571

General Resource TME Child

0572

General Resource TME Resource

0573

General Resource TME Group

0574

General Resource TME Role

0580

General Resource KERB Data

0590

General Resource PROXY Data

05A0

General Resource EIM Data

05B0

General Resource Alias Data

05C0

General Resource CDTINFO Data

05D0

General Resource ICTX Data

05E0

General Resource CFDEF Data

05F0

General Resource SIGVER Data

05G0

General Resource ICSF

05G1

General Resource ICSF Key Label

05G2

General Resource ICSF Certificate Identifier

05H0

General Resource MFA Factor Definition Record

05I0

General Resource MFA Policy Definition Record

05I1

General Resource MFA Policy Factors Record

05J1

General Resource CSDATA Record

05L0

General Resource JES Data Record

1210

User MFA Factor Tags Data Record

1560

General Resource Certificate Information

Format of the record type identification number

The identification number for the record type is in the format **PPSF**, where

PP

Profile type

01

For groups

02

For users

04

For data sets

05

For general resources

1nFor extension records, where *n* indicates profile type: 1, 2, 4, or 5, as listed above.

For example, record number 1560 identifies the General Resource Certificate Data extension record.

S

Segment number

0

Base segment

all others

Segment value as determined by the position of the segment in the template

F

Additional record qualifier

For example, when the profile type is 01 - 05, the additional record qualifier identifies a repeat group within the segment. A zero (0) indicates a non-repeat group within the segment.

The relationships among unloaded database records

The following figures describe how the records produced by the database unload utility relate to each other. The conventions used in the figures are:

- Only fields showing a relationship to another record type are described.
- A line shows a relationship between different types of records.
- The complete field names are in the format

prefix_fieldname

where *prefix* is the unique record prefix assigned to the record and *fieldname* identifies the field in the record. Each section provides the prefix added to the field names.

- The arrows on the connecting line clarify the relationship; they point to the field that had to have existed first in the RACF database.

For example, there is a user named GARREN. GARREN creates a group named TEST. The user ID named GARREN had to exist before the group TEST was created.

In terms of the output from database unload, there exists a user basic data record with GARREN in the USBD_NAME field. There also exists a group basic data record with TEST in the GPBD_NAME field and GARREN in the GPBD_OWNER_ID field.

The figures illustrating the relationships are located as follows:

- Group records, see [Figure 1 on page 361](#)
- User records, see [Figure 3 on page 363](#)
- Data set records, see [Figure 4 on page 364](#)
- General Resource records, see [Figure 5 on page 365](#).

Unloaded group record types

The prefix representing the record identifier is omitted in the pictorial diagrams. For group records, the prefixes are:

Record Name	Record Type	Record Prefix
Group Basic Data	0100	GPBD
Group Subgroups	0101	GPSGRP
Group Members	0102	GPMEM
Group Installation Data	0103	GPINSTD
Group DFP Data	0110	GPDFP
Group OMVS Data	0120	GPOMVS
Group OVM Data	0130	GPOVM
Group TME Data	0141	GPTME
Group CSDATA Custom Fields	0151	GPCSD

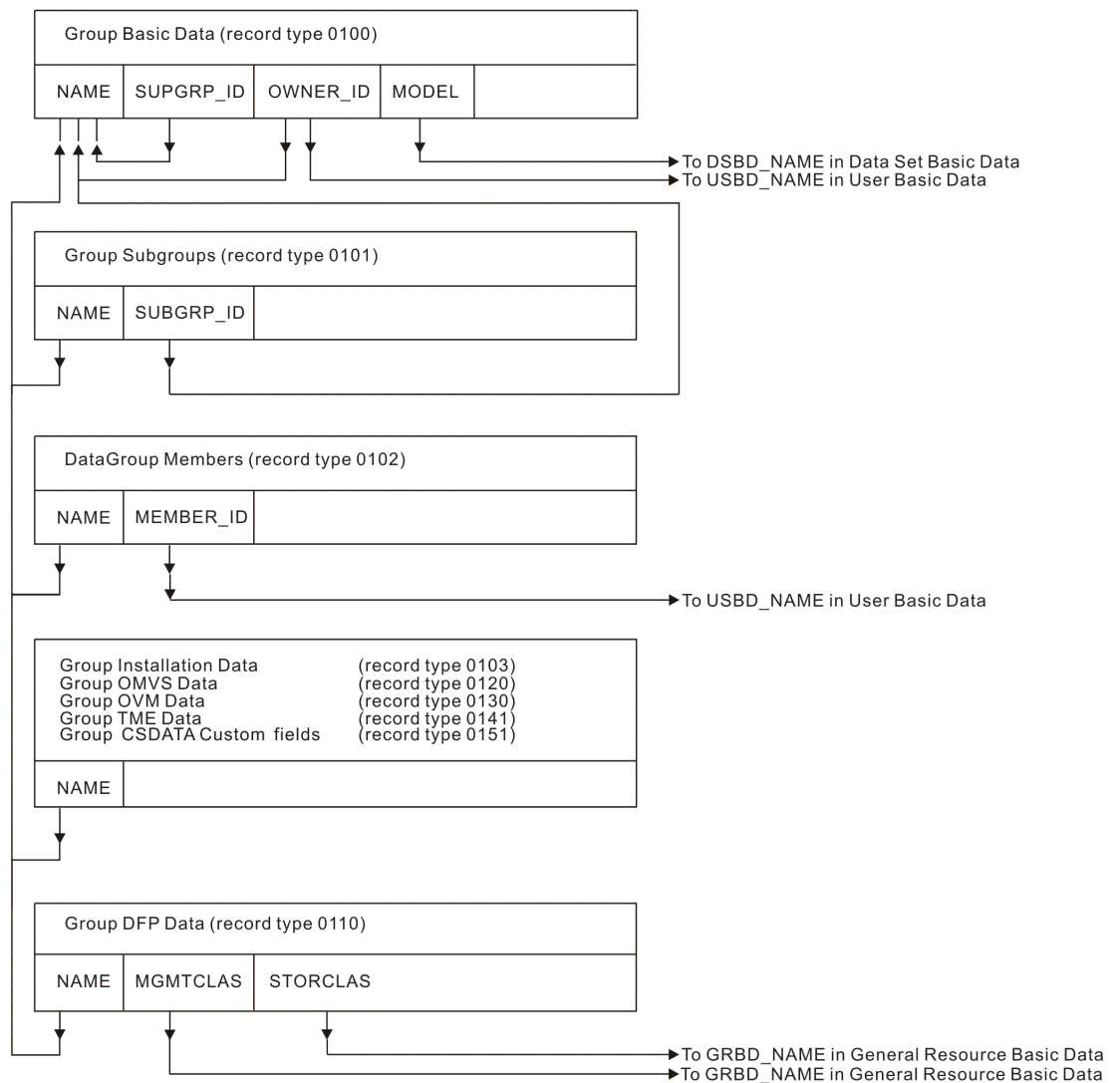


Figure 1. Relationship among the group record types

Unloaded user record types

The high level qualifier which represents the table identifier is omitted. For user records, these qualifiers are:

Record Name	Record Type	Record Prefix
User Basic Data	0200	USB
User Categories	0201	USCAT
User Classes	0202	USCLA
User Group Connections	0203	USGCON
User Installation Data	0204	USINST
User Connect Data	0205	USCON
User RRSF Data	0206	USRSF
User Certificate Data	0207	USCERT
User Mappings	0208	USNMAP
User Associated Distributed Mappings	0209	USDMAP
User MFA Factor Data Record	020A	USMFA
User MFA Factor Tags Data Record	020B	USMFAC
User DFP Data	0210	USDFP
User TSO Data	0220	USTSO
User CICS Data	0230	USCICS
User CICS Operator Classes	0231	USCOPC
User CICS RSL Keys	0232	USCRSL

User CICS TSL Keys	0233	USCTSL
User Language Data	0240	USLAN
User OPERPARM Data	0250	USOPR
User OPERPARM Scope	0251	USOPRP
User WORKATTR Data	0260	USWRK
User OMVS Data	0270	USOMVS
User NETVIEW Segment	0280	USNETV
User OPCLASS	0281	USNOPC
User DOMAINS	0282	USNDOM
User DCE Data	0290	USDCE
User OVM Data	02A0	USOVM
User LNOTES Data	02B0	USLNOT
User NDS Data	02C0	USNDS
User KERB Data	02D0	USKERB
User PROXY Data	02E0	USPROXY
User EIM Data	02F0	USEIM
User CSDATA Custom Fields	02G1	USCSD

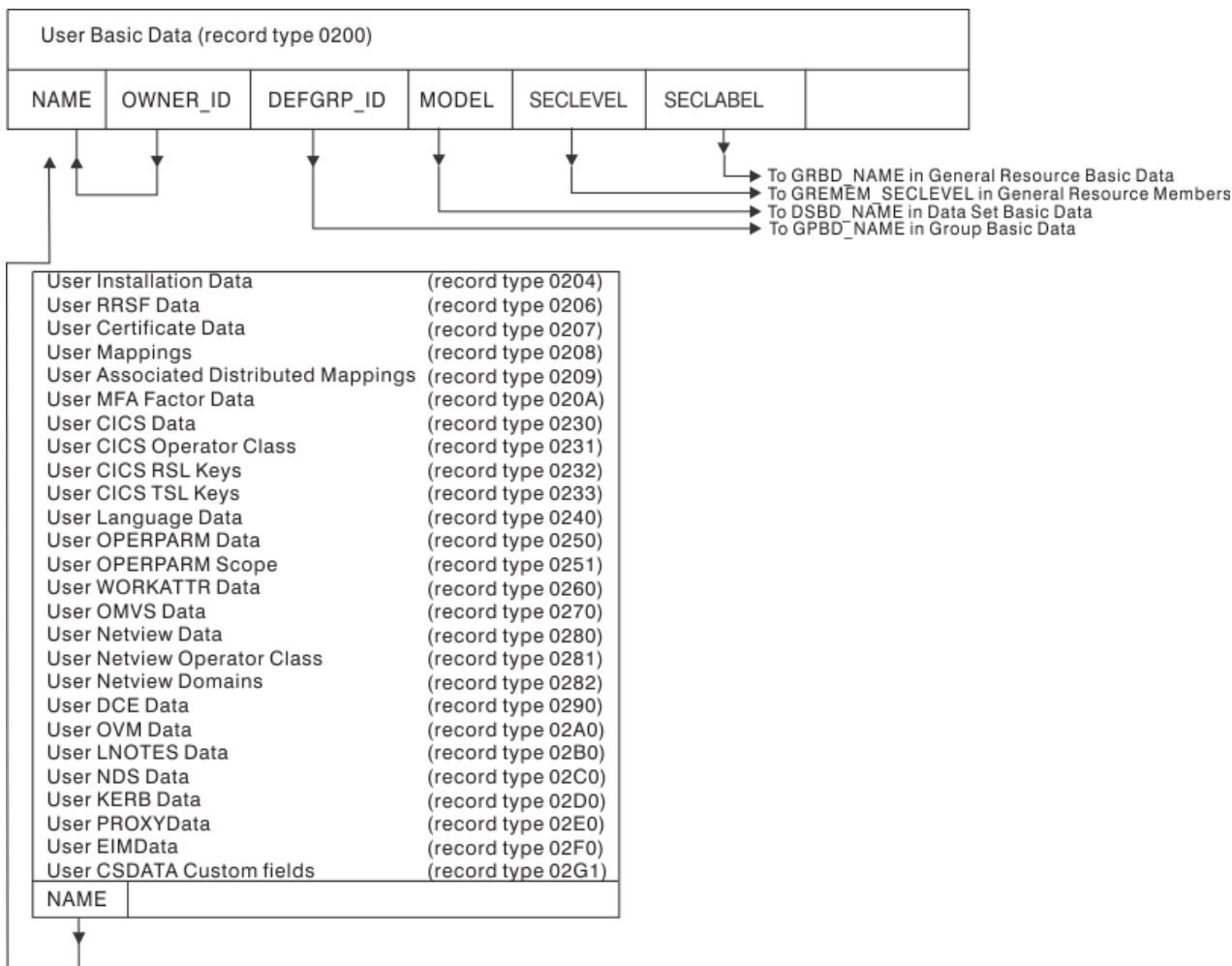


Figure 2. Relationship among the user record types (Part 1 of 2)

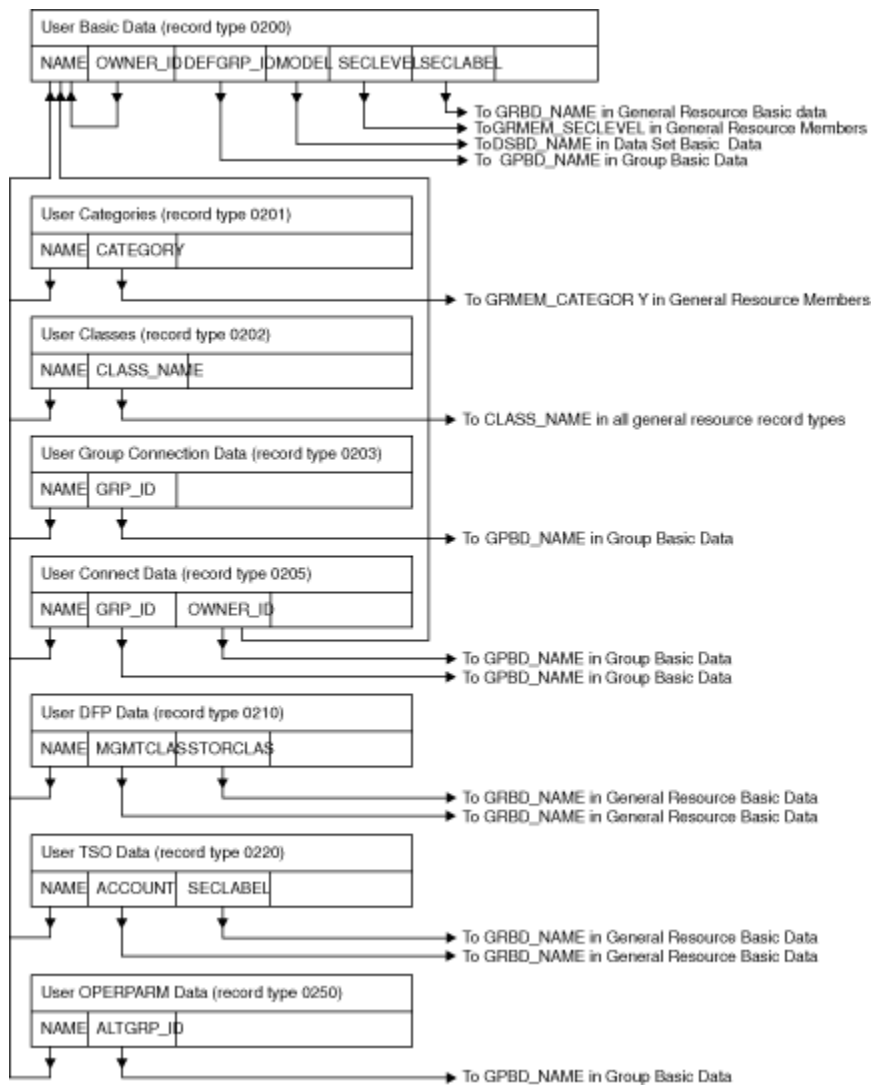


Figure 3. Relationship among the user record types (Part 2 of 2)

Unloaded data set record types

The high level qualifier which represents the table identifier is omitted. For data set records, these qualifiers are:

Record Name	Record Type	Record Prefix
Data Set Basic Data	0400	DSBD
Data Set Categories	0401	DSCAT
Data Set Conditional Access	0402	DSCACC
Data Set Volumes	0403	DSVOL
Data Set Access	0404	DSACC
Data Set Installation Data	0405	DSINSTD
Data Set DFP Data	0410	DSDFP
Data Set TME Role	0421	DSTME

The NAME/VOL field is a concatenation of the NAME field and VOLUME field.

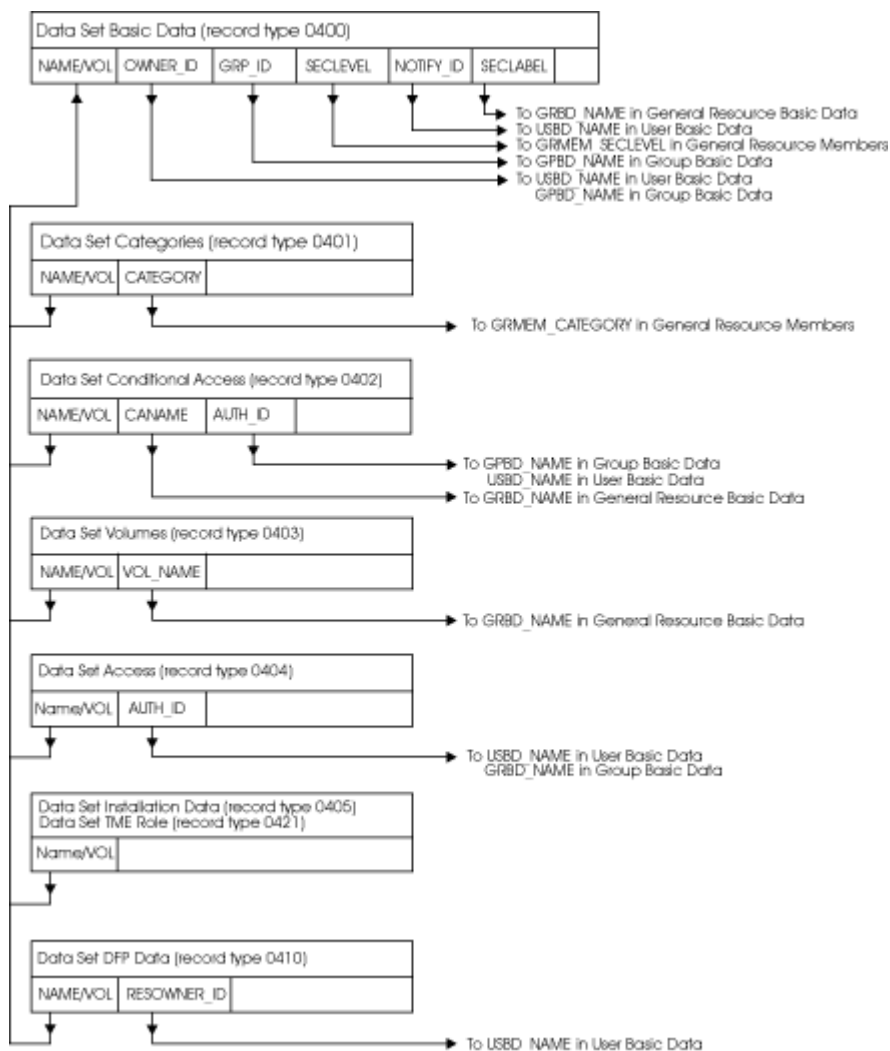


Figure 4. Relationship among the data set record types

Unloaded general resource record types

The high level qualifier which represents the table identifier is omitted. For general resource records, these qualifiers are:

Record Name	Record Type	Record Prefix
General Resource Basic Data	0500	GRBD
General Resource Tape Volume Data	0501	GRTVOL
General Resource Categories	0502	GRCAT
General Resource Members	0503	GRMEM
General Resource Volumes	0504	GRVOL
General Resource Access	0505	GRACC
General Resource Installation Data	0506	GRINSTD
General Resource Conditional Access	0507	GRCACC
General Filter Data	0508	GRFLTR
General Resource Distributed Identity Mapping Data	0509	GRDMAP
General Resource Session Data	0510	GRSES
General Resource Session Entities	0511	GRSESE
General Resource DLF Data	0520	GRDLF
General Resource DLF Job Names	0521	GRDLFJ
General Resource SSIGNON Data Record	0530	GRSIGN
General Resource Started Task Data	0540	GRST
General Resource SystemView Data	0550	GRSV
General Resource Certificate Data	0560	GRCERT
General Resource Certificate References	0561	CERTR
General Resource Key Ring Data	0562	KEYR
General Resource TME Data	0570	GRTME
General Resource TME Child	0571	GRTMEC
General Resource TME Resource	0572	GRTMER
General Resource TME Group	0573	GRTMEG

General Resource TME Role	0574	GRTMEE
General Resource KERB Data	0580	GRKERB
General Resource PROXY Data	0590	GRPROXY
General Resource EIM Data	05A0	GREIM
General Resource Alias Data	05B0	GRALIAS
General Resource CDTINFO Data	05C0	GRCDT
General Resource ICTX Data	05D0	GRICTX
General Resource CFDEF Data	05E0	GRCFDEF
General Resource SIGVER Data	05F0	GRSIG
General Resource ICSF	05G0	GRCSF
General Resource ICSF Key Label	05G1	GRCSFK
General Resource ICSF Certificate Identifier	05G2	GRCSFC
General Resource MFA Definition Record	05H0	GRMFA
General Resource JES Data Record	05L0	GRJES
General Resource Certificate Information	1560	CERTN

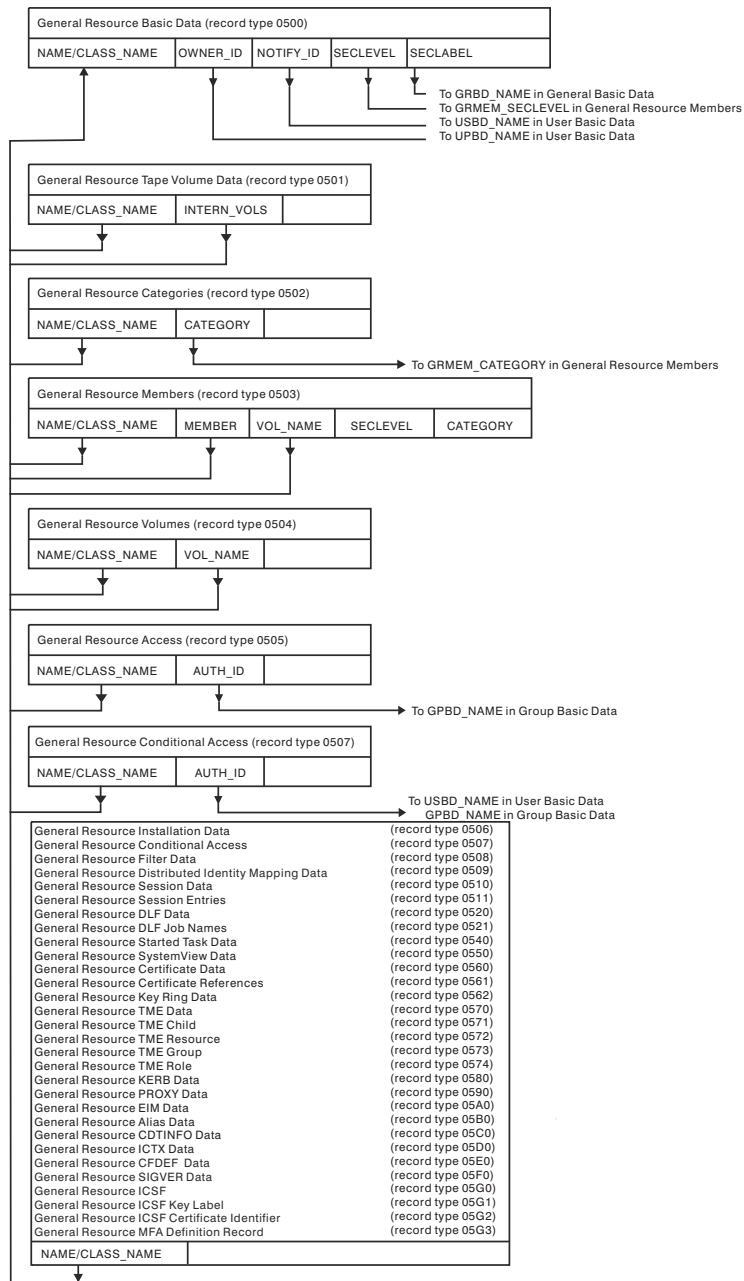


Figure 5. Relationship among the general resource record types

Conversion rules of the database unload utility

In unloading the database, these rules were followed:

- Each repeat group has its own record type.

For example, the repeat group representing the access list for data sets covered by a profile is ACL2CNT (the field name in the template). There is a data set access record (type 0404) created for each entry in the access list.

- Flag fields that are not mutually exclusive values (for example, 8-bit flags where more than one bit could be on at once) are defined as separate fields.

When this field is processed, it is unloaded as a 4-character field, with the values YES and NO as valid values. The field is left-justified.

- Flag fields that have mutually exclusive settings are unloaded as 8-character fields with a value corresponding to each bit setting.

For example, the UACC in a data set profile is a flag field in which each bit position corresponds to a universal access. The utility translates this single flag field into an 8-byte string with the value NONE, READ, UPDATE, CONTROL, or ALTER. If the flag field contains a value which is undefined, then the utility unloads the value as X<cc>, where cc is the hexadecimal value of the flag field.

- Encrypted and reserved fields are not unloaded.
- A maximum of 255 bytes are unloaded, except for the following fields:

Segment	Field	Bytes unloaded
PROXY	LDAP_HOST	1023
	BIND_DN	1023
EIM	DOMAIN_DN	1023
OMVS	HOME_PATH	1023
	PROGRAM	1023
OVM	HOME_PATH	1023
	PROGRAM	1023
	FSROOT	1023
DCE	DCE_NAME	1023
	HOMECELL	1023
CSDATA	All fields	Maximum available bytes are unloaded.

- Fields for the installation's data, such as INSTDATA or the USRxx fields, are unloaded without any decoding. The USRFLG field, however, is treated as a hexadecimal value and is represented by X<cc>.
- A single byte with the value blank (X'40') is placed between each field in the output record. This makes it easier to understand the output file when it is viewed.
- Fields in the database which contain null data have blanks unloaded, except for integer fields, which have a zero value unloaded. (Data is treated as null if 'FF' is coded as the default value for a character set in the base segment or if zeros are used in the character field in any segment other than the base segment.)
- Fields are converted to a readable form without interpretation of the current date or other information within the database.

For example, a user who shows as revoked when listed by LISTUSER, does not show as revoked with the raw UNLOAD data. If the revoked date is past, LISTUSER processes the data and shows the user as ATTRIBUTES=REVOKED, however the FLAG4 (USBD_REVOKE) bit in the unload data shows as NO.

Also, a protected user might show as N/A in the LISTUSER field for PASS-INTERVAL=, however the UNLOAD data might show a residual value in USBD_PWD_INTERVAL.

For more information, see [Comparing LISTUSER and LISTGRP output with IRRDBU00 in z/OS Security Server RACF Security Administrator's Guide](#).

Record formats produced by the database unload utility

The following topics contain a detailed description of the records that are produced by the database unload utility.

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

1. Descriptive name for the field
2. Type of field

Char

Character data.

Int

IntegerEBCDIC numeric data.

Time

A time value, in the form hh:mm:ss.

Date

A date value, in the form yyyy-mm-dd.

Yes/No

Flag data, having the value YES or NO.

3. Starting position for the field
4. Ending position for the field
5. Free form description of the field, which can contain the valid value constraints.

The complete record formats are located as follows:

- Group records, see [“Group record formats” on page 367](#)
- User records, see [“User record formats” on page 370](#)
- Data set records, see [“Data set record formats” on page 392](#)
- General Resource records, see [“General resource record formats” on page 397](#)

Note: For some applications, such as SQL/DS, the start and end positions must account for a 4-position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Group record formats

The records associated with groups are:

- Group Basic Data
- Group Subgroups
- Group Members
- Group Installation Data
- Group DFP Data
- Group OMVS Data
- Group OVM Data
- Group TME Role
- Group CSDATA Custom Fields

Group basic data record (0100)

The Group Basic Data record defines the basic information that defines a group. There is one record per group.

Table 196. Group Basic Data Record				
Field Name	Type	Position		Comments
		Start	End	
GPBD_RECORD_TYPE	Int	1	4	Record type of the Group Basic Data record (0100).
GPBD_NAME	Char	6	13	Group name as taken from the profile name.
GPBD_SUPGRP_ID	Char	15	22	Name of the superior group to this group.
GPBD_CREATE_DATE	Date	24	33	Date that the group was defined.
GPBD_OWNER_ID	Char	35	42	The user ID or group name which owns the profile.
GPBD_UACC	Char	44	51	The default universal access. Valid values are NONE for all groups other than the IBM-defined VSAMDSET group which has CREATE.
GPBD_NOTERMUACC	Char	53	56	Indicates if the group must be specifically authorized to use a particular terminal through the use of the PERMIT command. Valid Values include "Yes" and "No".
GPBD_INSTALL_DATA	Char	58	312	Installation-defined data.
GPBD_MODEL	Char	314	357	Data set profile that is used as a model for this group.
GPBD_UNIVERSAL	Char	359	362	Indicates if the group has the UNIVERSAL attribute. Valid Values include "Yes" and "No".

Group subgroups record (0101)

The Group Subgroups record defines the relationship between a group and any subgroups that are within the group. There is one record per group/subgroup combination.

Table 197. Group Subgroups Record				
Field Name	Type	Position		Comments
		Start	End	
GPSGRP_RECORD_TYPE	Int	1	4	Record type of the Group Subgroups record (0101).
GPSGRP_NAME	Char	6	13	Group name as taken from the profile name.
GPSGRP_SUBGRP_ID	Char	15	22	The name of a subgroup within the group.

Group members record (0102)

The Group Members record defines the relationship between a group and the members of the group. There is one record per group/member combination.

Table 198. Group Members Record				
Field Name	Type	Position		Comments
		Start	End	
GPMEM_RECORD_TYPE	Int	1	4	Record type of the Group Members record (0102).
GPMEM_NAME	Char	6	13	Group name as taken from the profile name.
GPMEM_MEMBER_ID	Char	15	22	A user ID within the group.
GPMEM_AUTH	Char	24	31	Indicates the authority that the user ID has within the group. Valid values are USE, CONNECT, JOIN, and CREATE.

Group installation data record (0103)

The Group Installation Data record defines the user data associated with a group.

This record type contains the data stored in the USRCNT repeat group, which is a field in the RACF database that is reserved for your installation's use. None of the RACF commands manipulate this field. Do not confuse this field with the GPBD_INSTALL_DATA field, shown in [Table 198 on page 368](#), which you enter into the database using the ADDGROUP and ALTGROUP commands.

There is one record per group/installation data combination.

Table 199. Group Installation Data Record				
Field Name	Type	Position		Comments
		Start	End	
GPINSTD_RECORD_TYPE	Int	1	4	Record type of the Group Installation Data record (0103).
GPINSTD_NAME	Char	6	13	Group name as taken from the profile name.
GPINSTD_USR_NAME	Char	15	22	The name of the installation-defined field.
GPINSTD_USR_DATA	Char	24	278	The data for the installation-defined field.
GPINSTD_USR_FLAG	Char	280	287	The flag for the installation-defined field in the form X<cc>.

Group DFP data record (0110)

The Group DFP Data record defines the information required by the System Managed Storage (SMS) facility of the Data Facility Product (DFP). The fields in these records define the characteristics of the data that this profile protects.

There is one record per group/DFP data combination.

Table 200. Group DFP Data Record				
Field Name	Type	Position		Comments
		Start	End	
GPDFP_RECORD_TYPE	Int	1	4	Record type of the Group DFP Data record (0110).
GPDFP_NAME	Char	6	13	Group name as taken from the profile name.
GPDFP_DATAAPPL	Char	15	22	Default application name for the group.
GPDFP_DATACLAS	Char	24	31	Default data class for the group.
GPDFP_MGMTCLAS	Char	33	40	Default management class for the group.
GPDFP_STORCLAS	Char	42	49	Default storage class for the group.

Group OMVS data record (0120)

The Group OMVS Data record defines the information required by z/OS UNIX to verify that users are associated with a valid z/OS UNIX group identifier (GID). These records define the GIDs that are assigned to RACF groups.

There is one record per group/GID combination.

Table 201. Group OMVS Data Record				
Field Name	Type	Position		Comments
		Start	End	
GPOMVS_RECORD_TYPE	Int	1	4	Record type of the Group OMVS Data record (0120).
GPOMVS_NAME	Char	6	13	Group name as taken from the profile name.
GPOMVS_GID	Char	15	24	OMVS z/OS UNIX group identifier (GID) associated with the group name from the profile.

Group OVM data record (0130)

The Group OVM Data record defines the OpenExtensions group identifiers (GIDs) that are assigned to RACF groups.

There is one record per group/GID combination.

Table 202. Group OVM Data Record				
Field Name	Type	Position		Comments
		Start	End	
GPOVM_RECORD_TYPE	Int	1	4	Record type of the Group OVM Data record (0130).
GPOVM_NAME	Char	6	13	Group name as taken from the profile name.
GPOVM_GID	Char	15	24	OpenExtensions group identifier (GID) associated with the group name from the profile.

Group TME role record (0141)

The Group TME Data record identifies ROLE profiles in which the group is referenced.

There is one record per group/role combination.

Table 203. Group TME Data Record				
Field Name	Type	Position		Comments
		Start	End	
GPTME_RECORD_TYPE	Int	1	4	Record type of the Group TME Data record (0141).
GPTME_NAME	Char	6	13	Group name as taken from the profile name.
GPTME_ROLE	Char	15	260	Role profile name.

Group CSDATA custom fields record (0151)

The Group CSDATA custom fields record defines the custom fields associated with a group. There is one record per combination of group and CSDATA custom fields.

Table 204. Group CSDATA custom fields record				
Field Name	Type	Position		Comments
		Start	End	
GPCSD_RECORD_TYPE	Int	1	4	Record type of the Group CSDATA custom fields (0151).
GPCSD_NAME	Char	6	13	Group name.
GPCSD_TYPE	Char	15	18	Data type for the custom field. Valid values are CHAR, FLAG, HEX, NUM.
GPCSD_KEY	Char	20	51	Custom field keyword; maximum length = 8.
GPCSD_VALUE	Char	53	1152	Custom field value.

User record formats

The records associated with users are:

- User Basic Data
- User Categories
- User Classes
- User Group Connections
- User Installation Data

- User Connect Data
- User RRSF Data
- User Certificate Name
- User Mappings
- User Associated Distributed Mappings
- User DFP Data
- User TSO Data
- User CICS Data
- User CICS Operator Classes
- User CICS RSL Keys
- User CICS TSL Keys
- User Language Data
- User OPERPARM Data
- User OPERPARM Scope
- User WORKATTR Data
- User OMVS Data
- User NETVIEW Segment
- User OPCLASS
- User DOMAINS
- User DCE Data
- User OVM Data
- User LNOTES Data
- User NDS Data
- User KERB Data
- User PROXY Data
- User EIM Data
- User CSDATA Custom Fields
- User MFA Data

User basic data record (0200)

The User Basic Data record defines the basic information about a user. There is one record per user.

Table 205. User basic data record				
Field Name	Type	Position		Comments
		Start	End	
USBD_RECORD_TYPE	Int	1	4	Record type of the User Basic Data record (0200).
USBD_NAME	Char	6	13	User ID as taken from the profile name.
USBD_CREATE_DATE	Date	15	24	The date that the profile was created.
USBD_OWNER_ID	Char	26	33	The user ID or group name that owns the profile.
USBD_ADSP	Char	35	38	Does the user have the ADSP attribute? Valid Values include "Yes" and "No".
USBD_SPECIAL	Char	40	43	Does the user have the SPECIAL attribute? Valid Values include "Yes" and "No".
USBD_OPER	Char	45	48	Does the user have the OPERATIONS attribute? Valid Values include "Yes" and "No".

Table 205. User basic data record (continued)				
Field Name	Type	Position		Comments
		Start	End	
USBD_REVOKE	Char	50	53	Is the user REVOKEd? Valid Values include "Yes" and "No".
USBD_GRPACC	Char	55	58	Does the user have the GRPACC attribute? Valid Values include "Yes" and "No".
USBD_PWD_INTERVAL	Int	60	62	The number of days that the user's password can be used.
USBD_PWD_DATE	Date	64	73	The date that the password was last changed.
USBD_PROGRAMMER	Char	75	94	The name associated with the user ID.
USBD_DEFGRP_ID	Char	96	103	The default group associated with the user.
USBD_LASTJOB_TIME	Time	105	112	The last recorded time that the user entered the system.
USBD_LASTJOB_DATE	Date	114	123	The last recorded date that the user entered the system.
USBD_INSTALL_DATA	Char	125	379	Installation-defined data.
USBD_UAUDIT	Char	381	384	Do all RACHECK and RACDEF SVCs cause logging? Valid Values include "Yes" and "No".
USBD_AUDITOR	Char	386	389	Does this user have the AUDITOR attribute? Valid Values include "Yes" and "No".
USBD_NOPWD	Char	391	394	<p>"YES" indicates that this user ID can log on without a password using OID card. "NO" indicates that this user must specify a password. "PRO" indicates a protected user ID. "PHR" indicates that the user has a password phrase.</p> <p>See also z/OS Security Server RACF Security Administrator's Guide.</p> <p>Note: USBD_PWD_ALG and USBD_PHR_ALG are the suggested fields to query to determine what combination of password and password phrase exists for a user.</p>
USBD_OIDCARD	Char	396	399	Does this user have OIDCARD data? Valid Values include "Yes" and "No".
USBD_PWD_GEN	Int	401	403	The current password generation number.
USBD_REVOKE_CNT	Int	405	407	The number of unsuccessful logon attempts.
USBD_MODEL	Char	409	452	The data set model profile name.
USBD_SECLLEVEL	Int	454	456	The user's security level.
USBD_REVOKE_DATE	Date	458	467	The date that the user is revoked.
USBD_RESUME_DATE	Date	469	478	The date that the user is resumed.
USBD_ACCESS_SUN	Char	480	483	Can the user access the system on Sunday? Valid Values include "Yes" and "No".
USBD_ACCESS_MON	Char	485	488	Can the user access the system on Monday? Valid Values include "Yes" and "No".
USBD_ACCESS_TUE	Char	490	493	Can the user access the system on Tuesday? Valid Values include "Yes" and "No".
USBD_ACCESS_WED	Char	495	498	Can the user access the system on Wednesday? Valid Values include "Yes" and "No".
USBD_ACCESS_THU	Char	500	503	Can the user access the system on Thursday? Valid Values include "Yes" and "No".
USBD_ACCESS_FRI	Char	505	508	Can the user access the system on Friday? Valid Values include "Yes" and "No".
USBD_ACCESS_SAT	Char	510	513	Can the user access the system on Saturday? Valid Values include "Yes" and "No".
USBD_START_TIME	Time	515	522	After what time can the user log on?

Table 205. User basic data record (continued)				
Field Name	Type	Position		Comments
		Start	End	
USBD_END_TIME	Time	524	531	After what time can the user not log on?
USBD_SECLABEL	Char	533	540	The user's default security label.
USBD_ATTRIBS	Char	542	549	Other user attributes (RSTD for users with RESTRICTED attribute).
USBD_PWDENV_EXISTS	Char	551	554	Has a PKCS#7 envelope been created for the user's current password? Valid Values include "Yes" and "No".
USBD_PWD_ASIS	Char	556	559	Should the password be evaluated in the case entered? Valid Values include "Yes" and "No".
USBD_PHR_DATE	Date	561	570	The date the password phrase was last changed.
USBD_PHR_GEN	Int	572	574	The current password phrase generation number.
USBD_CERT_SEQN	Int	576	585	Sequence number that is incremented whenever a certificate for the user is added, deleted, or altered. The starting value might not be 0.
USBD_PPHENV_EXISTS	Char	587	590	Has the user's current password phrase been PKCS#7 enveloped for possible retrieval? Valid Values include "Yes" and "No".
USBD_PWD_ALG	Char	592	603	Algorithm that is used to protect passwords. Possible values are "LEGACY", "KDFAES", and "NOPASSWORD".
USBD_LEG_PWDHIST_CT	Int	605	607	Number of legacy password history entries.
USBD_XPW_PWDHIST_CT	Int	609	611	Number of KDFAES password history entries.
USBD_PHR_ALG	Char	613	624	Algorithm that is used to protect password phrases. Possible values are "LEGACY", "KDFAES", and "NOPHRASE".
USBD_LEG_PHRHIST_CT	Int	626	628	Number of legacy password phrase history entries.
USBD_XPW_PHRHIST_CT	Int	630	632	Number of KDFAES password phrase history entries.
USBD_ROAUDIT	Char	634	637	This user can have a ROAUDIT attribute. Valid Values include "Yes" and "No".
USBD_MFA_FALLBACK	Char	639	641	This user can use a password or password phrase to logon to the system when MFA is unavailable. Valid Values include "Yes" and "No".
USBD_PHR_INTERVAL	Char	644	648	The number of days that the user's password phrase can be used. Note: Users without a password phrase interval will have the value 0. Users with a non-expiring password phrase interval (NOPHRASEINT) will have the value 65535 (X'FFFF').
USBD_CONTAIN	Char	650	653	Is the user contained (the user profile has the CONTAIN attribute). Valid values are "Yes" and "No."
USBD_NEVERCONTAIN	Char	655	658	Is the user restricted from being contained (the user profile cannot be given the CONTAIN attribute). Valid values are "Yes" and "No."

User categories record (0201)

The User Categories record defines the categories to which the user has access. There is one record per user/category combination.

Table 206. User Categories Record				
Field Name	Type	Position		Comments
		Start	End	
USCAT_RECORD_TYPE	Int	1	4	Record type of the User Categories record (0201).
USCAT_NAME	Char	6	13	User ID as taken from the profile name.
USCAT_CATEGORY	Int	15	19	Category to which the user has access.

User classes record (0202)

The User Classes record defines the classes in which the user can create profiles. There is one record per user/class combination.

Table 207. User Classes Record				
Field Name	Type	Position		Comments
		Start	End	
USCLA_RECORD_TYPE	Int	1	4	Record type of the User Classes record (0202).
USCLA_NAME	Char	6	13	User ID as taken from the profile name.
USCLA_CLASS	Char	15	22	A class in which the user is allowed to define profiles.

User group connections record (0203)

The User Group Connections record defines the groups with which the user is associated. There is one record per user connection.

Table 208. User Group Connections Record				
Field Name	Type	Position		Comments
		Start	End	
USGCON_RECORD_TYPE	Int	1	4	Record type of the User Group Connections record (0203).
USGCON_NAME	Char	6	13	User ID as taken from the profile name.
USGCON_GRP_ID	Char	15	22	The group with which the user is associated.

User installation data record (0204)

The User Installation Data record defines the user data associated with a user ID.

This record type contains the data stored in the USRCNT repeat group, which is a field in the RACF database that is reserved for your installation's use. None of the RACF commands manipulate this field. Do not confuse this field with the USER_INSTALL_DATA field, shown in [Table 205 on page 371](#), which you enter into the database using the ADDUSER and ALTUSER commands.

Table 209. User Installation Data Record				
Field Name	Type	Position		Comments
		Start	End	
USINSTD_RECORD_TYPE	Int	1	4	Record type of the User Installation Data record (0204).
USINSTD_NAME	Char	6	13	User ID as taken from the profile name.
USINSTD_USR_NAME	Char	15	22	The name of the installation-defined field.
USINSTD_USR_DATA	Char	24	278	The data for the installation-defined field.
USINSTD_USR_FLAG	Char	280	287	The flag for the installation-defined field in the form X<cc>.

User connect data record (0205)

The User Connect Data record defines the relationships between users and groups. There is one record per user connection.

Table 210. User Connect Data Record				
Field Name	Type	Position		Comments
		Start	End	
USCON_RECORD_TYPE	Int	1	4	Record type of the User Connect Data record (0205).
USCON_NAME	Char	6	13	User ID as taken from the profile name.
USCON_GRP_ID	Char	15	22	The group name.
USCON_CONNECT_DATE	Date	24	33	The date that the user was connected.
USCON_OWNER_ID	Char	35	42	The owner of the user-group connection.
USCON_LASTCON_TIME	Time	44	51	Time that the user last connected to this group.
USCON_LASTCON_DATE	Date	53	62	Date that the user last connected to this group.
USCON_UACC	Char	64	71	The default universal access authority for all new resources the user defines while connected to the specified group. Valid values are NONE, READ, UPDATE, CONTROL, and ALTER.
USCON_INIT_CNT	Int	73	77	The number of RACINITs issued for this user/group combination.
USCON_GRP_ADSP	Char	79	82	Does this user have the ADSP attribute in this group? Valid Values include "Yes" and "No".
USCON_GRP_SPECIAL	Char	84	87	Does this user have GROUP-SPECIAL in this group? Valid Values include "Yes" and "No".
USCON_GRP_OPER	Char	89	92	Does this user have GROUP-OPERATIONS in this group? Valid Values include "Yes" and "No".
USCON_REVOKE	Char	94	97	Is this user revoked? Valid Values include "Yes" and "No".
USCON_GRP_ACC	Char	99	102	Does this user have the GRPACC attribute? Valid Values include "Yes" and "No".
USCON_NOTERMUACC	Char	104	107	Does this user have the NOTERMUACC attribute in this group? Valid Values include "Yes" and "No".
USCON_GRP_AUDIT	Char	109	112	Does this user have the GROUP-AUDITOR attribute in this group? Valid Values include "Yes" and "No".
USCON_REVOKE_DATE	Date	114	123	The date that the user's connection to the group is revoked.
USCON_RESUME_DATE	Date	125	134	The date that the user's connection to the group is resumed.

User RRSF data record (0206)

The User RRSF data record defines the information required by RACF remote sharing facility (RRSF). There is one record per user/RRSF data combination.

Table 211. User RRSF Data Record				
Field Name	Type	Position		Comments
		Start	End	
USRSF_RECORD_TYPE	Int	1	4	Record type of the RRSF data record (0206).
USRSF_NAME	Char	6	13	User ID as taken from the profile name.
USRSF_TARG_NODE	Char	15	22	Target node name.
USRSF_TARG_USER_ID	Char	24	31	Target user ID.
USRSF_VERSION	Int	33	35	Version of this record.

Table 211. User RRSF Data Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
USRSF_PEER	Char	37	40	Is this a peer user ID? Valid Values include "Yes" and "No".
USRSF_MANAGING	Char	42	45	Is USRSF_NAME managing this ID? Valid Values include "Yes" and "No".
USRSF_MANAGED	Char	47	50	Is USRSF_NAME being managed by this ID? Valid Values include "Yes" and "No".
USRSF_REMOTE_PEND	Char	52	55	Is this remote RACF association pending? Valid Values include "Yes" and "No".
USRSF_LOCAL_PEND	Char	57	60	Is this local RACF association pending? Valid Values include "Yes" and "No".
USRSF_PWD_SYNC	Char	62	65	Is there password synchronization with this user ID? Valid Values include "Yes" and "No".
USRSF_REM_REFUSAL	Char	67	70	Was a system error encountered on the remote system? Valid Values include "Yes" and "No".
USRSF_DEFINE_DATE	Date	72	81	GMT date stamp for when this record was defined.
USRSF_DEFINE_TIME	Time	83	97	GMT time stamp for when this record was defined.
USRSF_ACCEPT_DATE	Date	99	108	GMT date stamp when this association was approved or refused. Based on the REMOTE_REFUSAL bit setting.
USRSF_ACCEPT_TIME	Time	110	124	GMT time stamp when this association was approved or refused. Based on the REMOTE_REFUSAL bit setting.
USRSF_CREATOR_ID	Char	126	133	User ID who created this entry.

User certificate name record (0207)

The User Certificate Name record defines the names of the certificate profiles in the DIGTCERT class that are associated with this user ID.

Note: RACF does not unload all fields in profiles in the DIGTCERT class. The digital certificate itself is not readable text and is the only field in the CERTDATA segment. Therefore, RACF bypasses the unloading of the CERTDATA segment of general resource profiles.

Table 212. User Certificate Name Record				
Field Name	Type	Position		Comments
		Start	End	
USCERT_RECORD_TYPE	Int	1	4	Record type of the user certificate name record (0207).
USCERT_NAME	Char	6	13	User ID as taken from the profile name.
USCERT_CERT_NAME	Char	15	260	Digital certificate name.
USCERT_CERTLABL	Char	262	293	Digital certificate label.

User associated mappings record (0208)

The User Associated Mappings Record defines the certificate name filter in the DIGTNMAP class associated with this user ID.

Table 213. User Associated Mappings Record				
Field Name	Type	Position		Comments
		Start	End	
USNMAP_RECORD_TYPE	Int	1	4	Record type of the User Associated Mappings record (0208).
USNMAP_NAME	Char	6	13	User ID as taken from the profile name.

Table 213. User Associated Mappings Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
USNMAP_LABEL	Char	15	46	The label associated with this mapping.
USNMAP_MAP_NAME	Char	48	293	The name of the DIGTNMAP profile associated with this user.

User associated distributed mappings record (0209)

The User Associated Distributed Mappings Record defines the IDIDMAP class profile name associated with this user ID.

Table 214. User Associated Distributed Mappings Record				
Field Name	Type	Position		Comments
		Start	End	
USDMAP_RECORD_TYPE	Int	1	4	Record type of the User Associated Distributed Mappings record (0209).
USDMAP_NAME	Char	6	13	User ID as taken from the profile name.
USDMAP_LABEL	Char	15	46	The label associated with this mapping.
USDMAP_MAP_NAME	Char	48	293	The name of the IDIDMAP profile associated with this user. Note: This value is stored in the RACF database in UTF-8 format. If possible, database unload changes this value to the EBCDIC format. If not possible, hexadecimal values are produced.

User MFA factor data record (020A)

The User MFA factor data record defines the basic information about a the MFA factor data.

Table 215. User MFA factor data record				
Field Name	Type	Position		Comments
		Start	End	
USMFA_RECORD_TYPE	Int	1	4	Record type of the user Multifactor authentication data record (020A).
USMFA_NAME	Char	6	13	User ID as taken from the profile name.
USMFA_FACTOR_NAME	Char	15	34	Factor name.
USMFA_FACTOR_ACTIVE	Date	36	54	Factor active date. Will be blank if factor is not ACTIVE.

User MFA policies record (020B)

The user MFA policies record (020B)

Table 216. User MFA policies record				
Field Name	Type	Position		Comments
		Start	End	
USMPOL_RECORD_TYPE	Int	1	4	Record type of the user Multi-factor authentication policies record (020B)
USMPOL_NAME	Char	6	13	User ID as taken from the profile name.
USMPOL_POLICY_NAME	Char	15	34	MFA Policy name.

User DFP data record (0210)

The User DFP Data record defines the information required by the System Managed Storage facility of the Data Facility Product (DFP). The fields in these records define the characteristics of the data that are created by the user. There is one record per user/DFP data combination.

Table 217. User DFP Data Record				
Field Name	Type	Position		Comments
		Start	End	
USDFP_RECORD_TYPE	Int	1	4	Record type of the User DFP data record (0210).
USDFP_NAME	Char	6	13	User ID as taken from the profile name.
USDFP_DATAAPPL	Char	15	22	Default application name for the user.
USDFP_DATACLAS	Char	24	31	Default data class for the user.
USDFP_MGMTCLAS	Char	33	40	Default management class for the user.
USDFP_STORCLAS	Char	42	49	Default storage class for the user.

User TSO data record (0220)

The user TSO data record defines the information required by TSO/E. There is one record per TSO user.

Table 218. User TSO data record				
Field Name	Type	Position		Comments
		Start	End	
USTSO_RECORD_TYPE	Int	1	4	Record type of the User TSO Data record (0220).
USTSO_NAME	Char	6	13	User ID as taken from the profile name.
USTSO_ACCOUNT	Char	15	54	The default account number.
USTSO_COMMAND	Char	56	135	The command issued at LOGON.
USTSO_DEST	Char	137	144	The default destination identifier.
USTSO_HOLD_CLASS	Char	146	146	The default hold class.
USTSO_JOB_CLASS	Char	148	148	The default job class.
USTSO_LOGON_PROC	Char	150	157	The default logon procedure.
USTSO_LOGON_SIZE	Int	159	168	The default logon region size.
USTSO_MSG_CLASS	Char	170	170	The default message class.
USTSO_LOGON_MAX	Int	172	181	The maximum logon region size.
USTSO_PERF_GROUP	Int	183	192	The performance group associated with the user.
USTSO_SYSOUT_CLASS	Char	194	194	The default sysout class.
USTSO_USER_DATA	Char	196	203	The TSO user data, in hexadecimal in the form X<cccc>.
USTSO_UNIT_NAME	Char	205	212	The default SYSDA device.
USTSO_SECLABEL	Char	214	221	The default logon security label.

User CICS data record (0230)

The User CICS data record defines the data required by the Customer Information Control System (CICS). There is one record per user/CICS data combination.

Table 219. User CICS data record

Field Name	Type	Position		Comments
		Start	End	
USCICS_RECORD_TYPE	Int	1	4	Record type of the User CICS Data record (0230).
USCICS_NAME	Char	6	13	User ID as taken from the profile name.
USCICS_OPIDENT	Char	15	17	The CICS operator identifier.
USCICS_OPPTY	Int	19	23	The CICS operator priority.
USCICS_NOFORCE	Char	25	28	Is the extended recovery facility (XRF) NOFORCE option in effect? Valid Values include "Yes" and "No".
USCICS_TIMEOUT	Char	30	34	The terminal time-out value. Expressed in hh:mm

User CICS operator classes record (0231)

The user CICS operator classes record defines the classes associated with a CICS operator. There is one record per user/CICS operator class combination.

Table 220. User CICS operator class record

Field Name	Type	Position		Comments
		Start	End	
USCOPC_RECORD_TYPE	Int	1	4	Record type of the User CICS Operator Class record (0231).
USCOPC_NAME	Char	6	13	User ID as taken from the profile name.
USCOPC_OPCLASS	Char	15	17	The class associated with the CICS operator.

User CICS RSL keys record (0232)

The user CICS RSL keys record defines the resource security level (RSL) keys associated with a CICS user. There is one record per combination of user and CICS RSL key.

Table 221. User CICS RSL key record

Field Name	Type	Position		Comments
		Start	End	
USCRSL_RECORD_TYPE	Int	1	4	Record type of the User CICS RSL keys record (0232).
USCRSL_NAME	Char	6	13	User ID as taken from the profile name.
USCRSL_KEY	Int	15	19	RSL key number.

User CICS TSL keys record (0233)

The User CICS TSL keys record defines the transaction security level (TSL) keys for a CICS user. There is one record per combination of user and CICS TSL key.

Table 222. User CICS TSL Key Record

Field Name	Type	Position		Comments
		Start	End	
USCTSL_RECORD_TYPE	Int	1	4	Record type of the User CICS TSL keys record (0233).
USCTSL_NAME	Char	6	13	User ID as taken from the profile name.
USCTSL_KEY	Int	15	19	TSL key number.

User language data record (0240)

The user language data record defines the primary and default languages for the user. There is one record per user/language combination.

Table 223. User language data record				
Field Name	Type	Position		Comments
		Start	End	
USLAN_RECORD_TYPE	Int	1	4	Record type of the User Language Data record (0240).
USLAN_NAME	Char	6	13	User ID as taken from the profile name.
USLAN_PRIMARY	Char	15	17	The primary language for the user.
USLAN_SECONDARY	Char	19	21	The secondary language for the user.

User OPERPARM data record (0250)

The user OPERPARM data record defines the operator characteristics for the user. There is one record per user/OPERPARM data combination.

Table 224. User OPERPARM data record				
Field Name	Type	Position		Comments
		Start	End	
USOPR_RECORD_TYPE	Int	1	4	Record type of the User OPERPARM Data record (0250).
USOPR_NAME	Char	6	13	User ID as taken from the profile name.
USOPR_STORAGE	Int	15	19	The number of megabytes of storage that can be used for message queuing.
USOPR_MASTERAUTH	Char	21	24	Does this user have MASTER console authority? Valid Values include "Yes" and "No".
USOPR_ALLAUTH	Char	26	29	Does this user have ALL console authority? Valid Values include "Yes" and "No".
USOPR_SYSAUTH	Char	31	34	Does this user have SYSAUTH console authority? Valid Values include "Yes" and "No".
USOPR_IOAUTH	Char	36	39	Does this user have I/O console authority? Valid Values include "Yes" and "No".
USOPR_CONSAUTH	Char	41	44	Does this user have CONS console authority? Valid Values include "Yes" and "No".
USOPR_INFOAUTH	Char	46	49	Does this user have INFO console authority? Valid Values include "Yes" and "No".
USOPR_TIMESTAMP	Char	51	54	Do console messages contain a timestamp? Valid Values include "Yes" and "No".
USOPR_SYSTEMID	Char	56	59	Do console messages contain a system ID? Valid Values include "Yes" and "No".
USOPR_JOBID	Char	61	64	Do console messages contain a job ID? Valid Values include "Yes" and "No".
USOPR_MSGID	Char	66	69	Do console messages contain a message ID? Valid Values include "Yes" and "No".
USOPR_X	Char	71	74	Are the job name and system name to be suppressed for messages issued from the JES3 global processor? Valid Values include "Yes" and "No".
USOPR_WTOR	Char	76	79	Does the console receive WTOR messages? Valid Values include "Yes" and "No".
USOPR_IMMEDIATE	Char	81	84	Does the console receive <i>immediate</i> messages? Valid Values include "Yes" and "No".

Table 224. User OPERPARM data record (continued)				
Field Name	Type	Position		Comments
		Start	End	
USOPR_CRITICAL	Char	86	89	Does the console receive <i>critical event</i> messages? Valid Values include "Yes" and "No".
USOPR_EVENTUAL	Char	91	94	Does the console receive <i>eventual event</i> messages? Valid Values include "Yes" and "No".
USOPR_INFO	Char	96	99	Does the console receive <i>informational</i> messages? Valid Values include "Yes" and "No".
USOPR_NOBROADCAST	Char	101	104	Are broadcast messages to this console suppressed? Valid Values include "Yes" and "No".
USOPR_ALL	Char	106	109	Does the console receive all messages? Valid Values include "Yes" and "No".
USOPR_JOBNAME	Char	111	114	Are job names monitored? Valid Values include "Yes" and "No".
USOPR_JOBNAMEST	Char	116	119	Are job names monitored with timestamps displayed? Valid Values include "Yes" and "No".
USOPR_SESS	Char	121	124	Are user IDs displayed with each TSO initiation and termination? Valid Values include "Yes" and "No".
USOPR_SESST	Char	126	129	Are user IDs and timestamps displayed with each TSO initiation and termination? Valid Values include "Yes" and "No".
USOPR_STATUS	Char	131	134	Are data set names and dispositions displayed with each data set that is freed? Valid Values include "Yes" and "No".
USOPR_ROUTECODE001	Char	136	139	Is this console enabled for route code 001? Valid Values include "Yes" and "No".
USOPR_ROUTECODE002	Char	141	144	Is this console enabled for route code 002? Valid Values include "Yes" and "No".
USOPR_ROUTECODE003	Char	146	149	Is this console enabled for route code 003? Valid Values include "Yes" and "No".
USOPR_ROUTECODE004	Char	151	154	Is this console enabled for route code 004? Valid Values include "Yes" and "No".
USOPR_ROUTECODE005	Char	156	159	Is this console enabled for route code 005? Valid Values include "Yes" and "No".
USOPR_ROUTECODE006	Char	161	164	Is this console enabled for route code 006? Valid Values include "Yes" and "No".
USOPR_ROUTECODE007	Char	166	169	Is this console enabled for route code 007? Valid Values include "Yes" and "No".
USOPR_ROUTECODE008	Char	171	174	Is this console enabled for route code 008? Valid Values include "Yes" and "No".
USOPR_ROUTECODE009	Char	176	179	Is this console enabled for route code 009? Valid Values include "Yes" and "No".
USOPR_ROUTECODE010	Char	181	184	Is this console enabled for route code 010? Valid Values include "Yes" and "No".
USOPR_ROUTECODE011	Char	186	189	Is this console enabled for route code 011? Valid Values include "Yes" and "No".
USOPR_ROUTECODE012	Char	191	194	Is this console enabled for route code 012? Valid Values include "Yes" and "No".
USOPR_ROUTECODE013	Char	196	199	Is this console enabled for route code 013? Valid Values include "Yes" and "No".
USOPR_ROUTECODE014	Char	201	204	Is this console enabled for route code 014? Valid Values include "Yes" and "No".

Table 224. User OPERPARM data record (continued)				
Field Name	Type	Position		Comments
		Start	End	
USOPR_ROUTECODE015	Char	206	209	Is this console enabled for route code 015? Valid Values include "Yes" and "No".
USOPR_ROUTECODE016	Char	211	214	Is this console enabled for route code 016? Valid Values include "Yes" and "No".
USOPR_ROUTECODE017	Char	216	219	Is this console enabled for route code 017? Valid Values include "Yes" and "No".
USOPR_ROUTECODE018	Char	221	224	Is this console enabled for route code 018? Valid Values include "Yes" and "No".
USOPR_ROUTECODE019	Char	226	229	Is this console enabled for route code 019? Valid Values include "Yes" and "No".
USOPR_ROUTECODE020	Char	231	234	Is this console enabled for route code 020? Valid Values include "Yes" and "No".
USOPR_ROUTECODE021	Char	236	239	Is this console enabled for route code 021? Valid Values include "Yes" and "No".
USOPR_ROUTECODE022	Char	241	244	Is this console enabled for route code 022? Valid Values include "Yes" and "No".
USOPR_ROUTECODE023	Char	246	249	Is this console enabled for route code 023? Valid Values include "Yes" and "No".
USOPR_ROUTECODE024	Char	251	254	Is this console enabled for route code 024? Valid Values include "Yes" and "No".
USOPR_ROUTECODE025	Char	256	259	Is this console enabled for route code 025? Valid Values include "Yes" and "No".
USOPR_ROUTECODE026	Char	261	264	Is this console enabled for route code 026? Valid Values include "Yes" and "No".
USOPR_ROUTECODE027	Char	266	269	Is this console enabled for route code 027? Valid Values include "Yes" and "No".
USOPR_ROUTECODE028	Char	271	274	Is this console enabled for route code 028? Valid Values include "Yes" and "No".
USOPR_ROUTECODE029	Char	276	279	Is this console enabled for route code 029? Valid Values include "Yes" and "No".
USOPR_ROUTECODE030	Char	281	284	Is this console enabled for route code 030? Valid Values include "Yes" and "No".
USOPR_ROUTECODE031	Char	286	289	Is this console enabled for route code 031? Valid Values include "Yes" and "No".
USOPR_ROUTECODE032	Char	291	294	Is this console enabled for route code 032? Valid Values include "Yes" and "No".
USOPR_ROUTECODE033	Char	296	299	Is this console enabled for route code 033? Valid Values include "Yes" and "No".
USOPR_ROUTECODE034	Char	301	304	Is this console enabled for route code 034? Valid Values include "Yes" and "No".
USOPR_ROUTECODE035	Char	306	309	Is this console enabled for route code 035? Valid Values include "Yes" and "No".
USOPR_ROUTECODE036	Char	311	314	Is this console enabled for route code 036? Valid Values include "Yes" and "No".
USOPR_ROUTECODE037	Char	316	319	Is this console enabled for route code 037? Valid Values include "Yes" and "No".
USOPR_ROUTECODE038	Char	321	324	Is this console enabled for route code 038? Valid Values include "Yes" and "No".

Table 224. User OPERPARM data record (continued)				
Field Name	Type	Position		Comments
		Start	End	
USOPR_ROUTECODE039	Char	326	329	Is this console enabled for route code 039? Valid Values include "Yes" and "No".
USOPR_ROUTECODE040	Char	331	334	Is this console enabled for route code 040? Valid Values include "Yes" and "No".
USOPR_ROUTECODE041	Char	336	339	Is this console enabled for route code 041? Valid Values include "Yes" and "No".
USOPR_ROUTECODE042	Char	341	344	Is this console enabled for route code 042? Valid Values include "Yes" and "No".
USOPR_ROUTECODE043	Char	346	349	Is this console enabled for route code 043? Valid Values include "Yes" and "No".
USOPR_ROUTECODE044	Char	351	354	Is this console enabled for route code 044? Valid Values include "Yes" and "No".
USOPR_ROUTECODE045	Char	356	359	Is this console enabled for route code 045? Valid Values include "Yes" and "No".
USOPR_ROUTECODE046	Char	361	364	Is this console enabled for route code 046? Valid Values include "Yes" and "No".
USOPR_ROUTECODE047	Char	366	369	Is this console enabled for route code 047? Valid Values include "Yes" and "No".
USOPR_ROUTECODE048	Char	371	374	Is this console enabled for route code 048? Valid Values include "Yes" and "No".
USOPR_ROUTECODE049	Char	376	379	Is this console enabled for route code 049? Valid Values include "Yes" and "No".
USOPR_ROUTECODE050	Char	381	384	Is this console enabled for route code 050? Valid Values include "Yes" and "No".
USOPR_ROUTECODE051	Char	386	389	Is this console enabled for route code 051? Valid Values include "Yes" and "No".
USOPR_ROUTECODE052	Char	391	394	Is this console enabled for route code 052? Valid Values include "Yes" and "No".
USOPR_ROUTECODE053	Char	396	399	Is this console enabled for route code 053? Valid Values include "Yes" and "No".
USOPR_ROUTECODE054	Char	401	404	Is this console enabled for route code 054? Valid Values include "Yes" and "No".
USOPR_ROUTECODE055	Char	406	409	Is this console enabled for route code 055? Valid Values include "Yes" and "No".
USOPR_ROUTECODE056	Char	411	414	Is this console enabled for route code 056? Valid Values include "Yes" and "No".
USOPR_ROUTECODE057	Char	416	419	Is this console enabled for route code 057? Valid Values include "Yes" and "No".
USOPR_ROUTECODE058	Char	421	424	Is this console enabled for route code 058? Valid Values include "Yes" and "No".
USOPR_ROUTECODE059	Char	426	429	Is this console enabled for route code 059? Valid Values include "Yes" and "No".
USOPR_ROUTECODE060	Char	431	434	Is this console enabled for route code 060? Valid Values include "Yes" and "No".
USOPR_ROUTECODE061	Char	436	439	Is this console enabled for route code 061? Valid Values include "Yes" and "No".
USOPR_ROUTECODE062	Char	441	444	Is this console enabled for route code 062? Valid Values include "Yes" and "No".

Table 224. User OPERPARM data record (continued)				
Field Name	Type	Position		Comments
		Start	End	
USOPR_ROUTECODE063	Char	446	449	Is this console enabled for route code 063? Valid Values include "Yes" and "No".
USOPR_ROUTECODE064	Char	451	454	Is this console enabled for route code 064? Valid Values include "Yes" and "No".
USOPR_ROUTECODE065	Char	456	459	Is this console enabled for route code 065? Valid Values include "Yes" and "No".
USOPR_ROUTECODE066	Char	461	464	Is this console enabled for route code 066? Valid Values include "Yes" and "No".
USOPR_ROUTECODE067	Char	466	469	Is this console enabled for route code 067? Valid Values include "Yes" and "No".
USOPR_ROUTECODE068	Char	471	474	Is this console enabled for route code 068? Valid Values include "Yes" and "No".
USOPR_ROUTECODE069	Char	476	479	Is this console enabled for route code 069? Valid Values include "Yes" and "No".
USOPR_ROUTECODE070	Char	481	484	Is this console enabled for route code 070? Valid Values include "Yes" and "No".
USOPR_ROUTECODE071	Char	486	489	Is this console enabled for route code 071? Valid Values include "Yes" and "No".
USOPR_ROUTECODE072	Char	491	494	Is this console enabled for route code 072? Valid Values include "Yes" and "No".
USOPR_ROUTECODE073	Char	496	499	Is this console enabled for route code 073? Valid Values include "Yes" and "No".
USOPR_ROUTECODE074	Char	501	504	Is this console enabled for route code 074? Valid Values include "Yes" and "No".
USOPR_ROUTECODE075	Char	506	509	Is this console enabled for route code 075? Valid Values include "Yes" and "No".
USOPR_ROUTECODE076	Char	511	514	Is this console enabled for route code 076? Valid Values include "Yes" and "No".
USOPR_ROUTECODE077	Char	516	519	Is this console enabled for route code 077? Valid Values include "Yes" and "No".
USOPR_ROUTECODE078	Char	521	524	Is this console enabled for route code 078? Valid Values include "Yes" and "No".
USOPR_ROUTECODE079	Char	526	529	Is this console enabled for route code 079? Valid Values include "Yes" and "No".
USOPR_ROUTECODE080	Char	531	534	Is this console enabled for route code 080? Valid Values include "Yes" and "No".
USOPR_ROUTECODE081	Char	536	539	Is this console enabled for route code 081? Valid Values include "Yes" and "No".
USOPR_ROUTECODE082	Char	541	544	Is this console enabled for route code 082? Valid Values include "Yes" and "No".
USOPR_ROUTECODE083	Char	546	549	Is this console enabled for route code 083? Valid Values include "Yes" and "No".
USOPR_ROUTECODE084	Char	551	554	Is this console enabled for route code 084? Valid Values include "Yes" and "No".
USOPR_ROUTECODE085	Char	556	559	Is this console enabled for route code 085? Valid Values include "Yes" and "No".
USOPR_ROUTECODE086	Char	561	564	Is this console enabled for route code 086? Valid Values include "Yes" and "No".

Table 224. User OPERPARM data record (continued)				
Field Name	Type	Position		Comments
		Start	End	
USOPR_ROUTECODE087	Char	566	569	Is this console enabled for route code 087? Valid Values include "Yes" and "No".
USOPR_ROUTECODE088	Char	571	574	Is this console enabled for route code 088? Valid Values include "Yes" and "No".
USOPR_ROUTECODE089	Char	576	579	Is this console enabled for route code 089? Valid Values include "Yes" and "No".
USOPR_ROUTECODE090	Char	581	584	Is this console enabled for route code 090? Valid Values include "Yes" and "No".
USOPR_ROUTECODE091	Char	586	589	Is this console enabled for route code 091? Valid Values include "Yes" and "No".
USOPR_ROUTECODE092	Char	591	594	Is this console enabled for route code 092? Valid Values include "Yes" and "No".
USOPR_ROUTECODE093	Char	596	599	Is this console enabled for route code 093? Valid Values include "Yes" and "No".
USOPR_ROUTECODE094	Char	601	604	Is this console enabled for route code 094? Valid Values include "Yes" and "No".
USOPR_ROUTECODE095	Char	606	609	Is this console enabled for route code 095? Valid Values include "Yes" and "No".
USOPR_ROUTECODE096	Char	611	614	Is this console enabled for route code 096? Valid Values include "Yes" and "No".
USOPR_ROUTECODE097	Char	616	619	Is this console enabled for route code 097? Valid Values include "Yes" and "No".
USOPR_ROUTECODE098	Char	621	624	Is this console enabled for route code 098? Valid Values include "Yes" and "No".
USOPR_ROUTECODE099	Char	626	629	Is this console enabled for route code 099? Valid Values include "Yes" and "No".
USOPR_ROUTECODE100	Char	631	634	Is this console enabled for route code 100? Valid Values include "Yes" and "No".
USOPR_ROUTECODE101	Char	636	639	Is this console enabled for route code 101? Valid Values include "Yes" and "No".
USOPR_ROUTECODE102	Char	641	644	Is this console enabled for route code 102? Valid Values include "Yes" and "No".
USOPR_ROUTECODE103	Char	646	649	Is this console enabled for route code 103? Valid Values include "Yes" and "No".
USOPR_ROUTECODE104	Char	651	654	Is this console enabled for route code 104? Valid Values include "Yes" and "No".
USOPR_ROUTECODE105	Char	656	659	Is this console enabled for route code 105? Valid Values include "Yes" and "No".
USOPR_ROUTECODE106	Char	661	664	Is this console enabled for route code 106? Valid Values include "Yes" and "No".
USOPR_ROUTECODE107	Char	666	669	Is this console enabled for route code 107? Valid Values include "Yes" and "No".
USOPR_ROUTECODE108	Char	671	674	Is this console enabled for route code 108? Valid Values include "Yes" and "No".
USOPR_ROUTECODE109	Char	676	679	Is this console enabled for route code 109? Valid Values include "Yes" and "No".
USOPR_ROUTECODE110	Char	681	684	Is this console enabled for route code 110? Valid Values include "Yes" and "No".

Table 224. User OPERPARM data record (continued)				
Field Name	Type	Position		Comments
		Start	End	
USOPR_ROUTECODE111	Char	686	689	Is this console enabled for route code 111? Valid Values include "Yes" and "No".
USOPR_ROUTECODE112	Char	691	694	Is this console enabled for route code 112? Valid Values include "Yes" and "No".
USOPR_ROUTECODE113	Char	696	699	Is this console enabled for route code 113? Valid Values include "Yes" and "No".
USOPR_ROUTECODE114	Char	701	704	Is this console enabled for route code 114? Valid Values include "Yes" and "No".
USOPR_ROUTECODE115	Char	706	709	Is this console enabled for route code 115? Valid Values include "Yes" and "No".
USOPR_ROUTECODE116	Char	711	714	Is this console enabled for route code 116? Valid Values include "Yes" and "No".
USOPR_ROUTECODE117	Char	716	719	Is this console enabled for route code 117? Valid Values include "Yes" and "No".
USOPR_ROUTECODE118	Char	721	724	Is this console enabled for route code 118? Valid Values include "Yes" and "No".
USOPR_ROUTECODE119	Char	726	729	Is this console enabled for route code 119? Valid Values include "Yes" and "No".
USOPR_ROUTECODE120	Char	731	734	Is this console enabled for route code 120? Valid Values include "Yes" and "No".
USOPR_ROUTECODE121	Char	736	739	Is this console enabled for route code 121? Valid Values include "Yes" and "No".
USOPR_ROUTECODE122	Char	741	744	Is this console enabled for route code 122? Valid Values include "Yes" and "No".
USOPR_ROUTECODE123	Char	746	749	Is this console enabled for route code 123? Valid Values include "Yes" and "No".
USOPR_ROUTECODE124	Char	751	754	Is this console enabled for route code 124? Valid Values include "Yes" and "No".
USOPR_ROUTECODE125	Char	756	759	Is this console enabled for route code 125? Valid Values include "Yes" and "No".
USOPR_ROUTECODE126	Char	761	764	Is this console enabled for route code 126? Valid Values include "Yes" and "No".
USOPR_ROUTECODE127	Char	766	769	Is this console enabled for route code 127? Valid Values include "Yes" and "No".
USOPR_ROUTECODE128	Char	771	774	Is this console enabled for route code 128? Valid Values include "Yes" and "No".
USOPR_LOGCMDRESP	Char	776	783	Specifies the logging of command responses received by the extended operator. Valid values are SYSTEM, NO, and blank.
USOPR_MIGRATIONID	Char	785	788	Is this extended operator to receive a migration ID?
USOPR_DELOPERMSG	Char	790	797	Does this extended operator receive delete operator messages? Valid values are NORMAL, ALL, and NONE.
USOPR_RETRIEVE_KEY	Char	799	806	Specifies a retrieval key used for searching. A null value is indicated by NONE.
USOPR_CMDSYS	Char	808	815	The name of the system that the extended operator is connected to for command processing.
USOPR_UD	Char	817	820	Is this operator to receive undeliverable messages? Valid Values include "Yes" and "No".
USOPR_ALTGRP_ID	Char	822	829	The default group associated with this operator.

Table 224. User OPERPARM data record (continued)				
Field Name	Type	Position		Comments
		Start	End	
USOPR_AUTO	Char	831	834	Is this operator to receive messages automated within the sysplex? Valid Values include "Yes" and "No".
USOPR_HC	Char	836	839	Is this operator to receive messages that are directed to hardcopy? Valid Values include "Yes" and "No".
USOPR_INT	Char	841	844	Is this operator to receive messages that are directed to console ID zero? Valid Values include "Yes" and "No".
USOPR_UNKN	Char	846	849	Is this operator to receive messages which are directed to unknown console IDs? Valid Values include "Yes" and "No".

User OPERPARM scope (0251)

The user OPERPARM scope record defines the scope of the operator. There is one record per user/OPERPARM scope combination.

Table 225. User OPERPARM scope record				
Field Name	Type	Position		Comments
		Start	End	
USOPRP_RECORD_TYPE	Int	1	4	Record type of the User OPERPARM Scope record (0251).
USOPRP_NAME	Char	6	13	User ID as taken from the profile name.
USOPRP_SYSTEM	Char	15	22	System name.

User WORKATTR data record (0260)

The user WORKATTR data record defines the logistical information for the user. There is one record per user/WORKATTR data combination.

Table 226. User WORKATTR data record				
Field Name	Type	Position		Comments
		Start	End	
USWRK_RECORD_TYPE	Int	1	4	Record type of the User WORKATTR Data record (0260).
USWRK_NAME	Char	6	13	User ID as taken from the profile name.
USWRK_AREA_NAME	Char	15	74	Area for delivery.
USWRK_BUILDING	Char	76	135	Building for delivery.
USWRK_DEPARTMENT	Char	137	196	Department for delivery.
USWRK_ROOM	Char	198	257	Room for delivery.
USWRK_ADDR_LINE1	Char	259	318	Address line 1.
USWRK_ADDR_LINE2	Char	320	379	Address line 2.
USWRK_ADDR_LINE3	Char	381	440	Address line 3.
USWRK_ADDR_LINE4	Char	442	501	Address line 4.
USWRK_ACCOUNT	Char	503	757	Account number.
USWRK_EMAIL_ADDRESS	Char	759	1004	E-mail address.

User OMVS data record (0270)

The user OMVS data record defines the information required by z/OS UNIX to verify that users are associated with a valid z/OS UNIX user identifier (UID). These records define the UIDs that have been assigned to RACF users, their default directory, default program name, and user limits.

There is only one record per user/UID data combination.

Table 227. User OMVS data record				
Field Name	Type	Position		Comments
		Start	End	
USOMVS_RECORD_TYPE	Int	1	4	Record type of the User Data record (0270).
USOMVS_NAME	Char	6	13	User name as taken from the profile name.
USOMVS_UID	Char	15	24	z/OS UNIX user identifier (UID) associated with the user name from the profile.
USOMVS_HOME_PATH	Char	26	1048	HOME PATH associated with the z/OS UNIX user identifier (UID).
USOMVS_PROGRAM	Char	1050	2072	Default Program associated with the z/OS UNIX user identifier (UID).
USOMVS_CPUTIMEMAX	Int	2074	2083	Maximum CPU time associated with the UID.
USOMVS_ASSIZEMAX	Int	2085	2094	Maximum address space size associated with the UID.
USOMVS_FILEPROCMA	Int	2096	2105	Maximum active or open files associated with the UID.
USOMVS_PROCUSEMAX	Int	2107	2116	Maximum number of processes associated with the UID.
USOMVS_THREADSMA	Int	2118	2127	Maximum number of threads associated with the UID.
USOMVS_MMAPAREAMAX	Int	2129	2138	Maximum mappable storage amount associated with the UID.
USOMVS_MEMLIMIT	Char	2140	2148	Maximum size of non-shared memory
USOMVS_SHMEMMAX	Char	2150	2158	Maximum size of shared memory

User NETVIEW segment record (0280)

The user NETVIEW segment record defines the information required by NetView.

There is only one record per user profile that contains a NETVIEW segment.

Table 228. User NETVIEW segment record				
Field Name	Type	Position		Comments
		Start	End	
USNETV_RECORD_TYPE	Int	1	4	Record type of the user NETVIEW segment record (0280).
USNETV_NAME	Char	6	13	User ID as taken from profile name
USNETV_IC	Char	15	269	Command list processed at logon
USNETV_CONSNAME	Char	271	278	Default console name
USNETV_CTL	Char	280	287	CTL value: GENERAL, GLOBAL, or SPECIFIC
USNETV_MSGRECV	Char	289	292	Eligible to receive unsolicited messages? Valid Values include "Yes" and "No".
USNETV_NGMFADMN	Char	294	297	Authorized to NetView graphic monitoring facility? Valid Values include "Yes" and "No".
USNETV_NGMFVSPN	Char	299	306	Value of view span options

User OPCLASS record (0281)

The user OPCLASS record defines the information required by NetView.

There is only one record per OPCLASS specified in the NETVIEW segment.

Table 229. User OPCLASS record				
Field Name	Type	Position		Comments
		Start	End	
USNOPC_RECORD_TYPE	Int	1	4	Record type of the user OPCLASS record (0281).
USNOPC_NAME	Char	6	13	User ID as taken from the profile name
USNOPC_OPCLASS	Int	15	19	OPCLASS value from 1 to 2040

User DOMAINS record (0282)

The user DOMAINS record defines the information required by NetView.

There is only one record per DOMAIN specified in the NETVIEW segment.

Table 230. User DOMAINS record				
Field Name	Type	Position		Comments
		Start	End	
USNDOM_RECORD_TYPE	Int	1	4	Record type of the user DOMAINS record (0282).
USNDOM_NAME	Char	6	13	User ID as taken from the profile name
USNDOM_DOMAINS	Char	15	19	DOMAIN value.

User DCE data record (0290)

The user DCE data record defines the non-repeating group information that is contained within the user's DCE segment.

Table 231. User DCE data record				
Field Name	Type	Position		Comments
		Start	End	
USDCE_RECORD_TYPE	Int	1	4	Record type of the user DCE data record (0290).
USDCE_NAME	Char	6	13	RACF user name as taken from the profile name.
USDCE_UUID	Char	15	50	DCE UUID associated with the user name from the profile.
USDCE_DCE_NAME	Char	52	1074	DCE principal name associated with this user.
USDCE_HOMECELL	Char	1076	2098	Home cell name.
USDCE_HOMEUUID	Char	2100	2135	Home cell UUID.
USDCE_AUTOLOGIN	Char	2137	2140	Is this user eligible for an automatic DCE login? Valid Values include "Yes" and "No".

User OVM data record (02A0)

The user OVM data record defines the information required by OpenExtensions. These records define the user identifiers (UIDs) that have been assigned to RACF users, their default directory, default program name, and the file system root.

Table 232. User OVM data record				
Field Name	Type	Position		Comments
		Start	End	
USOVM_RECORD_TYPE	Int	1	4	Record type of the user OVM data record (02A0).
USOVM_NAME	Char	6	13	User name as taken from the profile name.

Table 232. User OVM data record (continued)				
Field Name	Type	Position		Comments
		Start	End	
USOVM_UID	Char	15	24	User identifier (UID) associated with the user name from the profile.
USOVM_HOME_PATH	Char	26	1048	Home path associated with the user identifier (UID).
USOVM_PROGRAM	Char	1050	2072	Default program associated with the user identifier (UID).
USOVM_FSROOT	Char	2074	3096	File system root for this user.

User LNOTES data record (02B0)

The user LNOTES data record contains the Lotus Notes for z/OS information defined in the LNOTES segment of the user's profile.

Table 233. User LNOTES data record				
Field Name	Type	Position		Comments
		Start	End	
USLNOT_RECORD_TYPE	Int	1	4	Record type of the LNOTES data record (02B0).
USLNOT_NAME	Char	6	13	User ID as taken from the profile name.
USLNOT_SNAME	Char	15	78	LNOTES short name associated with the user ID.

User NDS data record (02C0)

The user NDS data record contains the Novell Directory Services for OS/390 information defined in the NDS segment of the user's profile.

Table 234. User NDS data record				
Field Name	Type	Position		Comments
		Start	End	
USNDS_RECORD_TYPE	Int	1	4	Record type of the NDS data record (02C0).
USNDS_NAME	Char	6	13	User ID as taken from the profile name.
USNDS_UNAME	Char	15	260	NDS user name associated with the user ID.

User KERB data record (02D0)

The user KERB data record defines the Kerberos principal information for a user. There is one record per user profile that contains a KERB segment.

Table 235. User KERB data record				
Field Name	Type	Position		Comments
		Start	End	
USKERB_RECORD_TYPE	Int	1	4	Record type of the User KERB segment record (02D0).
USKERB_NAME	Char	6	13	RACF user name as taken from the profile.
USKERB_KERBNAME	Char	15	254	The Kerberos principal name.
USKERB_MAX_LIFE	Int	256	265	Maximum ticket life.
USKERB_KEY_VERS	Int	267	269	Current key version.
USKERB_ENCRYPT_DES	Char	271	274	Is key encryption using DES enabled? Valid Values include "Yes" and "No".

Table 235. User KERB data record (continued)				
Field Name	Type	Position		Comments
		Start	End	
USKERB_ENCRYPT_DES3	Char	276	279	Is key encryption using DES3 enabled? Valid Values include "Yes" and "No".
USKERB_ENCRYPT_DESD	Char	281	284	Is key encryption using DES with derivation enabled? Valid Values include "Yes" and "No".
USKERB_ENCRPT_A128	Char	286	289	Is key encryption using AES128 enabled? Valid Values include "Yes" and "No".
USKERB_ENCRPT_A256	Char	291	294	Is key encryption using AES256 enabled? Valid Values include "Yes" and "No".
USKERB_ENCRPT_A128SHA2	Char	296	299	Is key encryption using AES128 SHA2 enabled? Valid Values include "Yes" and "No".
USKERB_ENCRPT_A256SHA2	Char	301	304	Is key encryption using AES256 SHA2 enabled? Valid Values include "Yes" and "No".
USKERB_KEY_FROM	Char	351	358	Key source. Valid values are PASSWORD or PHRASE.

User PROXY record (02E0)

The user PROXY record identifies default information related to the LDAP proxy for a user. There is only one record per user profile that contains a PROXY segment.

Table 236. User PROXY record				
Field Name	Type	Position		Comments
		Start	End	
USPROXY_RECORD_TYPE	Int	1	4	Record type of the user PROXY record (02E0).
USPROXY_NAME	Char	6	13	RACF user name as taken from the profile name.
USPROXY_LDAP_HOST	Char	15	1037	LDAP server URL.
USPROXY_BIND_DN	Char	1039	2061	LDAP BIND distinguished name.

User EIM data record (02F0)

The user EIM record defines the LDAPBIND profile for a user. There is one record per user profile that contains the EIM segment.

Table 237. User EIM Record				
Field Name	Type	Position		Comments
		Start	End	
USEIM_RECORD_TYPE	Int	1	4	Record type of the user EIM segment record (02F0).
USEIM_NAME	Char	6	13	User name.
USEIM_LDAPPROF	Char	15	260	EIM LDAPBIND profile name.

User CSDATA custom fields record (02G1)

The User CSDATA custom fields record defines custom fields associated with a user. There is one record per combination of user and CSDATA custom fields.

Table 238. User CSDATA Custom fields record				
Field Name	Type	Position		Comments
		Start	End	
USCSD_RECORD_TYPE	Int	1	4	Record type of the user CSDATA custom fields record (02G1).
USCSD_NAME	Char	6	13	User name.
USCSD_TYPE	Char	15	18	Data type for the custom field. Valid values are CHAR, FLAG, HEX, NUM.
USCSD_KEY	Char	20	51	Custom field keyword; maximum length = 8.
USCSD_VALUE	Char	53	1152	Custom field value.

User MFA factor tags data record (1210)

The User MFA factor tags data record defines the basic information about a the MFA factor tags data.

Table 239. User MFA factor tags data record				
Field Name	Type	Position		Comments
		Start	End	
USMFAC_RECORD_TYPE	Int	1	4	Record type of the user Multifactor authentication factor configuration data record (1210).
USMFAC_NAME	Char	6	13	User ID as taken from the profile name.
USMFAC_FACTOR_NAME	Char	15	34	Factor name.
USMFAC_TAG_NAME	Char	36	55	The tag name associated with the factor.
USMFAC_TAG_VALUE	Char	57	1080	Tag value associated with the tag name.

Data set record formats

The records associated with data sets are:

- Data Set Basic Data
- Data Set Categories
- Data Set Conditional Access
- Data Set Volumes
- Data Set Access
- Data Set Installation Data
- Data Set DFP Data
- Data Set TME Data
- Data set CSDATA Record

Data set basic data record (0400)

The Data Set Basic Data record defines the basic information for a data set. There is one record per data set profile.

Table 240. Data Set Basic Data Record				
Field Name	Type	Position		Comments
		Start	End	
DSBD_RECORD_TYPE	Int	1	4	Record type of the Data Set Basic Data record (0400).
DSBD_NAME	Char	6	49	Data set name as taken from the profile name.

Table 240. Data Set Basic Data Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
DSBD_VOL	Char	51	56	Volume upon which this data set resides. Blank if the profile is generic, and *MODEL if the profile is a model profile.
DSBD_GENERIC	Yes/No	58	61	Is this a generic profile?
DSBD_CREATE_DATE	Date	63	72	Date the profile was created.
DSBD_OWNER_ID	Char	74	81	The user ID or group name that owns the profile.
DSBD_LASTREF_DATE	Date	83	92	The date that the data set was last referenced.
DSBD_LASTCHG_DATE	Date	94	103	The date that the data set was last changed.
DSBD_ALTER_CNT	Int	105	109	The number of times that the data set was accessed with ALTER authority.
DSBD_CONTROL_CNT	Int	111	115	The number of times that the data set was accessed with CONTROL authority.
DSBD_UPDATE_CNT	Int	117	121	The number of times that the data set was accessed with UPDATE authority.
DSBD_READ_CNT	Int	123	127	The number of times that the data set was accessed with READ authority.
DSBD_UACC	Char	129	136	The universal access of this data set. Valid values are NONE, EXECUTE, READ, UPDATE, CONTROL, and ALTER.
DSBD_GRPDS	Yes/No	138	141	Is this a group data set?
DSBD_AUDIT_LEVEL	Char	143	150	Indicates the level of resource-owner-specified auditing that is performed. Valid values are ALL, SUCCESS, FAIL, and NONE.
DSBD_GRP_ID	Char	152	159	The connect group of the user who created this data set.
DSBD_DS_TYPE	Char	161	168	The type of the data set. Valid values are VSAM, NONVSAM, TAPE, and MODEL.
DSBD_LEVEL	Int	170	172	The level of the data set.
DSBD_DEVICE_NAME	Char	174	181	The EBCDIC name of the device type on which the data set resides.
DSBD_GAUDIT_LEVEL	Char	183	190	Indicates the level of auditor-specified auditing that is performed. Valid values are ALL, SUCCESS, FAIL, and NONE.
DSBD_INSTALL_DATA	Char	192	446	Installation-defined data.
DSBD_AUDIT_OKQUAL	Char	448	455	The resource-owner-specified successful access audit qualifier. This is set to blanks if AUDIT_LEVEL is NONE. Otherwise, it is set to either READ, UPDATE, CONTROL, or ALTER.
DSBD_AUDIT_FAQUAL	Char	457	464	The resource-owner-specified failing access audit qualifier. This is set to blanks if AUDIT_LEVEL is NONE. Otherwise, it is set to either READ, UPDATE, CONTROL, or ALTER.
DSBD_GAUDIT_OKQUAL	Char	466	473	The auditor-specified successful access audit qualifier. This is set to blanks if GAUDIT_LEVEL is NONE. Otherwise, it is set to either READ, UPDATE, CONTROL, or ALTER.
DSBD_GAUDIT_FAQUAL	Char	475	482	The auditor-specified failing access audit qualifier. This is set to blanks if GAUDIT_LEVEL is NONE. Otherwise, it is set to either READ, UPDATE, CONTROL, or ALTER.
DSBD_WARNING	Yes/No	484	487	Does this data set have the WARNING attribute?
DSBD_SECLEVEL	Int	489	491	The data set security level.
DSBD_NOTIFY_ID	Char	493	500	User ID that is notified when violations occur.
DSBD_RETENTION	Int	502	506	Retention period of the data set.

Table 240. Data Set Basic Data Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
DSBD_ERASE	Yes/No	508	511	For a DASD data set, is this data set scratched when the data set is deleted?
DSBD_SECLABEL	Char	513	520	Security label of the data set.
DSBD_RESERVED_01	Char	522	526	Reserved
DSBD_RESERVED_02	Char	528	532	Reserved

Data set categories record (0401)

The Data Set Categories record defines the categories to which a data set belongs. There is one record per data set/category combination.

Table 241. Data Set Categories Record				
Field Name	Type	Position		Comments
		Start	End	
DSCAT_RECORD_TYPE	Int	1	4	Record type of the Data Set Categories record (0401).
DSCAT_NAME	Char	6	49	Data set name as taken from the profile name.
DSCAT_VOL	Char	51	56	Volume upon which this data set resides. Blank if the profile is generic, and *MODEL if the profile is a model profile.
DSCAT_CATEGORY	Int	58	62	Category associated with this data set.

Data set conditional access record (0402)

The Data Set Conditional Access record defines the data sets that have conditional access permissions. There is one record per data set/access combination.

Table 242. Data Set Conditional Access Record				
Field Name	Type	Position		Comments
		Start	End	
DSCACC_RECORD_TYPE	Int	1	4	Record type of the Data Set Conditional Access record (0402).
DSCACC_NAME	Char	6	49	Data set name as taken from the profile name.
DSCACC_VOL	Char	51	56	Volume upon which this data set resides. Blank if the profile is generic, and *MODEL if the profile is a model profile.
DSCACC_CATYPE	Char	58	65	The type of conditional access checking that is being performed. Valid values are APPCPORT, PROGRAM, CONSOLE, TERMINAL, JESINPUT, and SERVAUTH.
DSCACC_CANAME	Char	67	74	The name of a conditional access element that is permitted access.
DSCACC_AUTH_ID	Char	76	83	The user ID or group name that is authorized to the data set.
DSCACC_ACCESS	Char	85	92	The access of the conditional access element/user combination. Valid values are NONE, EXECUTE, READ, UPDATE, CONTROL, and ALTER.
DSCACC_ACCESS_CNT	Int	94	98	The number of times that the data set was accessed.
DSCACC_NET_ID	Char	100	107	The network name when DSCACC_CATYPE is APPCPORT.
DSCACC_CACRITERIA	Char	109	352	The IP name when DSCACC_CATYPE is SERVAUTH.

Data set volumes record (0403)

The Data Set Volumes record defines the volumes upon which a data set resides. There is one record per data set/volume combination. Records exist in this table only for discrete data set profiles.

Table 243. Data Set Volumes Record				
Field Name	Type	Position		Comments
		Start	End	
DSVOL_RECORD_TYPE	Int	1	4	Record type of the Data Set Volumes record (0403).
DSVOL_NAME	Char	6	49	Data set name as taken from the profile name.
DSVOL_VOL	Char	51	56	Volume upon which this data set resides.
DSVOL_VOL_NAME	Char	58	63	A volume upon which the data set resides.

Data set access record (0404)

The Data Set Access record defines the users or groups that are allowed to access data. There is one record per data set/authorization combination.

Table 244. Data Set Access Record				
Field Name	Type	Position		Comments
		Start	End	
DSACC_RECORD_TYPE	Int	1	4	Record type of the Data Set Access Record (0404).
DSACC_NAME	Char	6	49	Data set name as taken from the profile name.
DSACC_VOL	Char	51	56	Volume upon which this data set resides. Blank if the profile is generic, and *MODEL if the profile is a model profile.
DSACC_AUTH_ID	Char	58	65	The user ID or group name that is authorized to the data set.
DSACC_ACCESS	Char	67	74	The access allowed to the user. Valid values are NONE, EXECUTE, READ, UPDATE, CONTROL, and ALTER.
DSACC_ACCESS_CNT	Int	76	80	The number of times that the data set was accessed.

Data set installation data record (0405)

The Data Set Installation Data record defines the user data that is associated with a data set profile. There is one record per data set/installation data combination.

This record type contains the data stored in the USRCNT repeat group, which is a field in the RACF database that is reserved for your installation's use. None of the RACF commands manipulate this field. Do not confuse this field with the DSBD_INSTALL_DATA field, shown in [Table 240 on page 392](#), which you enter into the database using the ADDSD and ALTDSD commands.

Table 245. Data Set Installation Data Record				
Field Name	Type	Position		Comments
		Start	End	
DSINSTD_RECORD_TYPE	Int	1	4	Record type of the Data Set Installation Data Record (0405).
DSINSTD_NAME	Char	6	49	Data set name as taken from the profile name.
DSINSTD_VOL	Char	51	56	Volume upon which this data set resides. Blank if the profile is generic, and *MODEL if the profile is a model profile.
DSINSTD_USR_NAME	Char	58	65	The name of the installation-defined field.
DSINSTD_USR_DATA	Char	67	321	The data for the installation-defined field.
DSINSTD_USR_FLAG	Char	323	330	The flag for the installation-defined field in the form X<cc>.

Data set member record (0406)

The Data Set Member record defines the member data that is associated with a data set profile. There is one record per data set/installation data combination.

Table 246. Data Set Member Record				
Field Name	Type	Position		Comments
		Start	End	
DSMEM_RECORD_TYPE	Int	1	4	Record type of the Data Set Member Data Record (0406).
DSMEM_NAME	Char	6	49	Data set name as taken from the profile name.
DSMEM_VOL	Char	51	56	Volume upon which this data set resides. Blank if the profile is generic, and *MODEL if the profile is a model profile.
DSMEM_MEMBER_NAME	Char	58	65	Member name.
DSMEM_AUTH_ID	Char	67	74	The user ID or group name that is authorized to the member.
DSMEM_ACCESS	Char	76	83	The access that is allowed to the ID. Valid values are "NONE", "READ", "UPDATE", and "CONTROL".

Data set DFP data record (0410)

The Data Set DFP Data record defines the DFP information required by the System Managed Storage (SMS) facility of the Data Facility Product (DFP). There is one record per data set/DFP data combination.

Table 247. Data Set DFP Data Record				
Field Name	Type	Position		Comments
		Start	End	
DSDFP_RECORD_TYPE	Int	1	4	Record type of the Data Set DFP Data record (0410).
DSDFP_NAME	Char	6	49	Data set name as taken from the profile name.
DSDFP_VOL	Char	51	56	Volume upon which this data set resides. Blank if the profile is generic, and *MODEL if the profile is a model profile.
DSDFP_RESOWNER_ID	Char	58	65	The resource owner of the data set.
DSDFP_DATAKEY	Char	67	130	The label of the ICSF that is used to encrypt the data of any newly allocated data set.
DSDFP_INCLUDE_TAPE	Char	132	135	Is data set encryption allowed for tape data sets covered by this profile? Valid values include YES, NO, and blanks. Blanks indicate that SMS decides.
DSDFP_INCLUDE_PDSE	Char	137	140	Is data set encryption allowed for tape data sets covered by this profile? Valid values include YES, NO, and blanks. Blanks indicate that SMS decides.
DSDFP_INCLUDE_SEQ	Char	142	145	Is data set encryption allowed for tape data sets covered by this profile? Valid values include YES, NO, and blanks. Blanks indicate that SMS decides.
DSDFP_RESERVED_01		147	150	Reserved - blanks
DSDFP_RESERVED_02		152	155	Reserved - blanks
DSDFP_RESERVED_03		157	160	Reserved - blanks
DSDFP_RESERVED_04		162	165	Reserved - blanks
DSDFP_RESERVED_05		167	170	Reserved - blanks
DSDFP_RESERVED_06		172	175	Reserved - blanks
DSDFP_RESERVED_07		177	180	Reserved - blanks
DSDFP_RESERVED_08		182	185	Reserved - blanks
DSDFP_RESERVED_09		187	190	Reserved - blanks

Table 247. Data Set DFP Data Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
DSDFP_RESERVED_10		192	195	Reserved - blanks
DSDFP_RESERVED_11		197	200	Reserved - blanks
DSDFP_RESERVED_12		202	205	Reserved - blanks
DSDFP_RESERVED_13		207	210	Reserved - blanks

Data set TME role record (0421)

The Data Set TME role record identifies ROLE profiles and access authorities referencing the data set. There is one record per data set/role combination.

Table 248. Data Set TME Data Record				
Field Name	Type	Position		Comments
		Start	End	
DSTME_RECORD_TYPE	Int	1	4	Record type of the Data Set TME Data Record (0421).
DSTME_NAME	Char	6	49	Data set name as taken from the profile name.
DSTME_VOL	Char	51	56	Volume upon which this data set resides. Blank if the profile is generic, and *MODEL if the profile is a model profile.
DSTME_ROLE_NAME	Char	58	303	Role profile name.
DSTME_ACCESS_AUTH	Char	305	312	Access permission to this resource as defined by the role.
DSTME_COND_CLASS	Char	314	321	Class name for conditional access.
DSTME_COND_PROF	Char	323	568	Resource profile for conditional access.

Data set CSDATA record (0431)

Table 249. Data set CSDATA record				
Field Name	Type	Position		Comments
		Start	End	
DSCSD_RECORD_TYPE	Int	1	4	Record type of the Data Set CSDATA custom fields record (0431).
DSCSD_NAME	Char	6	49	Data set name as taken from the profile name.
DSCSD_VOL	Char	51	56	Volume upon which this data set resides. Blank if the profile is generic, and *MODEL if the profile is a model profile.
DSCSD_TYPE	Char	58	61	Data type for the custom field. Valid values are CHAR, FLAG, HEX, NUM.
DSCSD_KEY	Char	63	94	Custom field keyword; maximum length = 8.
DSCSD_VALUE	Char	96	1195	Custom field value.

General resource record formats

The records associated with general resources are:

- General Resource Basic Data
- General Resource Tape Volume Data
- General Resource Categories
- General Resource Members
- General Resource Volumes

- General Resource Access
- General Resource Installation Data
- General Resource Conditional Access Data
- General Resource Filter Data
- General Resource Distributed Identity Mapping Data
- General Resource Session Data
- General Resource Session Entities
- General Resource DLF Data
- General Resource DLF Job Names
- General Resource Started Task Data
- General Resource SystemView Data
- General Resource Certificate Data
- General Resource Certificate Reference
- General Resource Key Ring Data
- General Resource TME Data
- General Resource TME Child
- General Resource TME Resource
- General Resource TME Group
- General Resource TME Role
- General Resource KERB Data
- General Resource PROXY Data
- General Resource EIM Data
- General Resource Alias Data
- General Resource CDTINFO Data
- General Resource ICTX Data
- General Resource CFDEF Data
- General Resource SIGVER Data
- General Resource ICSF
- General Resource ICSF Key Label
- General Resource ICSF Certificate Identifier
- General Resource MFA Factor Definition Record
- General Resource MFPOLICY Definition Record
- General Resource MFA Policy Factors Record
- General Resource CSDATA Record
- General Resource IDTPARMS Definition Record
- General Resource Certificate Information Record

Note: The digital certificates stored in the CERTDATA segment of general resource profiles are not readable text. Therefore, RACF bypasses the unload of the CERTDATA segment, and there is no record for this data.

General resource basic data record (0500)

The General Resource Basic Data record defines the basic information about a general resource. There is one record per general resource profile.

Table 250. General Resource Basic Data Record				
Field Name	Type	Position		Comments
		Start	End	
GRBD_RECORD_TYPE	Int	1	4	Record type of the General Resource Basic Data record (0500).
GRBD_NAME	Char	6	251	General resource name as taken from the profile name. Note: When GRBD_CLASS_NAME is IDIDMAP, this value is stored in the RACF database in UTF-8 format. If possible, database unload changes this value to the EBCDIC format. If not possible, hexadecimal values are produced.
GRBD_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRBD_GENERIC	Char	262	265	Is this a generic profile? Valid Values include "Yes" and "No".
GRBD_CLASS	Int	267	269	The class number of the profile.
GRBD_CREATE_DATE	Date	271	280	Date the profile was created.
GRBD_OWNER_ID	Char	282	289	The user ID or group name which owns the profile.
GRBD_LASTREF_DATE	Date	291	300	The date that the resource was last referenced.
GRBD_LASTCHG_DATE	Date	302	311	The date that the resource was last changed.
GRBD_ALTER_CNT	Int	313	317	The number of times that the resource was accessed with ALTER authority.
GRBD_CONTROL_CNT	Int	319	323	The number of times that the resource was accessed with CONTROL authority.
GRBD_UPDATE_CNT	Int	325	329	The number of times that the resource was accessed with UPDATE authority.
GRBD_READ_CNT	Int	331	335	The number of times that the resource was accessed with READ authority.
GRBD_UACC	Char	337	344	The universal access of this resource. For profiles in classes other than DIGTCERT, the valid values are NONE, READ, EXECUTE, UPDATE, CONTROL, and ALTER. For DIGTCERT profiles, the valid values are TRUST, NOTRUST, and HIGHTRST.
GRBD_AUDIT_LEVEL	Char	346	353	Indicates the level of resource-owner-specified auditing that is performed. Valid values are ALL, SUCCESS, FAIL, and NONE.
GRBD_LEVEL	Int	355	357	The level of the resource.
GRBD_GAUDIT_LEVEL	Char	359	366	Indicates the level of auditor-specified auditing that is performed. Valid values are ALL, SUCCESS, FAIL, and NONE.
GRBD_INSTALL_DATA	Char	368	622	Installation-defined data.
GRBD_AUDIT_OKQUAL	Char	624	631	The resource-owner-specified successful access audit qualifier. This is set to blanks if AUDIT_LEVEL is NONE. Otherwise, it is set to either READ, UPDATE, CONTROL, or ALTER.
GRBD_AUDIT_FAQUAL	Char	633	640	The resource-owner-specified failing access audit qualifier. This is set to blanks if AUDIT_LEVEL is NONE. Otherwise, it is set to either READ, UPDATE, CONTROL, or ALTER.
GRBD_GAUDIT_OKQUAL	Char	642	649	The auditor-specified successful access audit qualifier. This is set to blanks if GAUDIT_LEVEL is NONE. Otherwise, it is set to either READ, UPDATE, CONTROL, or ALTER.
GRBD_GAUDIT_FAQUAL	Char	651	658	The auditor-specified failing access audit qualifier. This is set to blanks if GAUDIT_LEVEL is NONE. Otherwise, it is set to either READ, UPDATE, CONTROL, or ALTER.
GRBD_WARNING	Char	660	663	Does this resource have the WARNING attribute? Valid Values include "Yes" and "No".

Table 250. General Resource Basic Data Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
GRBD_SINGLEDSD	Char	665	668	If this is a TAPEVOL profile, is there only one data set on this tape? Valid Values include "Yes" and "No".
GRBD_AUTO	Char	670	673	If this is a TAPEVOL profile, is the TAPEVOL protection automatic? Valid Values include "Yes" and "No".
GRBD_TVTOC	Char	675	678	If this is a TAPEVOL profile, is there a tape volume table of contents on this tape? Valid Values include "Yes" and "No".
GRBD_NOTIFY_ID	Char	680	687	User ID that is notified when violations occur.
GRBD_ACCESS_SUN	Char	689	692	Can the terminal be used on Sunday? Valid Values include "Yes" and "No".
GRBD_ACCESS_MON	Char	694	697	Can the terminal be used on Monday? Valid Values include "Yes" and "No".
GRBD_ACCESS_TUE	Char	699	702	Can the terminal be used on Tuesday? Valid Values include "Yes" and "No".
GRBD_ACCESS_WED	Char	704	707	Can the terminal be used on Wednesday? Valid Values include "Yes" and "No".
GRBD_ACCESS_THU	Char	709	712	Can the terminal be used on Thursday? Valid Values include "Yes" and "No".
GRBD_ACCESS_FRI	Char	714	717	Can the terminal be used on Friday? Valid Values include "Yes" and "No".
GRBD_ACCESS_SAT	Char	719	722	Can the terminal be used on Saturday? Valid Values include "Yes" and "No".
GRBD_START_TIME	Time	724	731	After what time can a user logon from this terminal?
GRBD_END_TIME	Time	733	740	After what time can a user not logon from this terminal?
GRBD_ZONE_OFFSET	Char	742	746	Time zone in which the terminal is located. Expressed as hh:mm. Blank if the time zone has not been specified.
GRBD_ZONE_DIRECT	Char	748	748	The direction of the time zone shift. Valid values are E(east), W(west), and blank.
GRBD_SECLEVEL	Int	750	752	The security level of the general resource.
GRBD_APPL_DATA	Char	754	1008	Installation-defined data.
GRBD_SECLABEL	Char	1010	1017	The security label for the general resource.

General resource tape volume data record (0501)

The General Resource Tape Volume Data Record defines the characteristics of the tape volume upon which a data set resides. There is one record per general resource/tape volume combination.

Table 251. General Resource Tape Volume Record				
Field Name	Type	Position		Comments
		Start	End	
GRTVOL_RECORD_TYPE	Int	1	4	Record type of the General Resource Tape Volume Data record (0501).
GRTVOL_NAME	Char	6	251	General resource name as taken from the profile name.
GRTVOL_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely TAPEVOL.
GRTVOL_SEQUENCE	Int	262	266	The file sequence number of the tape data set.
GRTVOL_CREATE_DATE	Date	268	277	Creation date of the tape data set.

Table 251. General Resource Tape Volume Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
GRTVOL_DISCRETE	Char	279	282	Does a discrete profile exist? Valid Values include "Yes" and "No".
GRTVOL_INTERN_NAME	Char	284	327	The RACF internal data set name.
GRTVOL_INTERN_VOLS	Char	329	583	The volumes upon which the data set resides.
GRTVOL_CREATE_NAME	Char	585	628	The data set name used when creating the data set.

General resource categories record (0502)

The General Resource Categories record defines the categories associated with a general resource. There is one record per general resource/category combination.

Table 252. General Resource Categories Record				
Field Name	Type	Position		Comments
		Start	End	
GRCAT_RECORD_TYPE	Int	1	4	Record type of the General Resources Categories record (0502).
GRCAT_NAME	Char	6	251	General resource name as taken from the profile name.
GRCAT_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRCAT_CATEGORY	Int	262	266	Category to which this general resource belongs.

General resource members record (0503)

The General Resource Members record defines the members of a general resource profile group. There is one record per general resource/member combination.

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

Table 253. General Resource Members Record				
Field Name	Type	Position		Comments
		Start	End	
GRMEM_RECORD_TYPE	Int	1	4	Record type of the General Resource Members record (0503).
GRMEM_NAME	Char	6	251	General resource name as taken from the profile name.
GRMEM_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.

Table 253. General Resource Members Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
GRMEM_MEMBER	Char	262	516	Member value for this general resource. <ul style="list-style-type: none"> For VMXEVENT profiles, this is the element that is being audited. For PROGRAM profiles, this is the name of the data set which contains the program. For GLOBAL profiles, this is the name of the resource for which a global access applies. For SECDATA security level (SECLEVEL) profiles, this is the level name. For SECDATA CATEGORY profiles, this is the category name. For NODES profiles, this is the user ID, group name, and security label translation data. For SECLABEL profiles, this is a 4-byte SMF ID.
GRMEM_GLOBAL_ACC	Char	518	525	If this is a GLOBAL profile, this is the access that is allowed. Valid values are NONE, READ, UPDATE, CONTROL, and ALTER.
GRMEM_PADS_DATA	Char	527	534	If this is a PROGRAM profile, this field contains the Program Access to Data Set (PADS) information for the profile. Valid values are PADCHK and NOPADCHK.
GRMEM_VOL_NAME	Char	536	541	If this is a PROGRAM profile, this field defines the volume upon which the program resides.
GRMEM_VMEVENT_DATA	Char	543	547	If this is a VMXEVENT profile, this field defines the level of auditing that is being performed. Valid values are CTL, AUDIT, and NOCTL.
GRMEM_SECLEVEL	Int	549	553	If this is a SECLEVEL profile in the SECDATA class, this is the numeric security level that is associated with the SECLEVEL.
GRMEM_CATEGORY	Int	555	559	If this is a CATEGORY profile in the SECDATA class, this is the numeric category that is associated with the CATEGORY.

General resource volumes record (0504)

The General Resource Volumes record defines the volumes in a tape volume set. There is one record per tape volume set/volume combination.

Table 254. General Resource Volumes Record				
Field Name	Type	Position		Comments
		Start	End	
GRVOL_RECORD_TYPE	Int	1	4	Record type of the General Resources Volumes record (0504).
GRVOL_NAME	Char	6	251	General resource name as taken from the profile name.
GRVOL_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely TAPEVOL.
GRVOL_VOL_NAME	Char	262	267	Name of a volume in a tape volume set.

General resource access record (0505)

The General Resource Access record defines the users or groups who have specific access to general resources. There is one record per general resource/authorization combination.

Table 255. General Resource Access Record				
Field Name	Type	Position		Comments
		Start	End	
GRACC_RECORD_TYPE	Int	1	4	Record type of the General Resource Access record (0505).
GRACC_NAME	Char	6	251	General resource name as taken from the profile name.
GRACC_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRACC_AUTH_ID	Char	262	269	User ID or group name which is authorized to use the general resource.
GRACC_ACCESS	Char	271	278	The authority that the user or group has over the resource. Valid values are NONE, EXECUTE, READ, UPDATE, CONTROL, and ALTER.
GRACC_ACCESS_CNT	Int	280	284	The number of times that the resource was accessed.

General resource installation data record (0506)

The General Resource Installation Data record defines the user data associated with a general resource. There is one record per general resource/data combination.

This record type contains data stored in the USRCNT repeat group, which is a field in the RACF database that is reserved for your installation's use. None of the RACF commands manipulate this field. Do not confuse this field with the GRBD_INSTALL_DATA field, shown in [Table 250 on page 399](#), which you enter into the database using the RDEFINE and RALTER commands.

Table 256. General Resource Installation Data Record				
Field Name	Type	Position		Comments
		Start	End	
GRINSTD_RECORD_TYPE	Int	1	4	Record type of the General Resource Installation Data record (0506).
GRINSTD_NAME	Char	6	251	General resource name as taken from the profile name.
GRINSTD_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRINSTD_USR_NAME	Char	262	269	The name of the installation-defined field.
GRINSTD_USR_DATA	Char	271	525	The data for the installation-defined field.
GRINSTD_USR_FLAG	Char	527	534	The flag for the installation-defined field in the form X<nn>.

General resource conditional access record (0507)

The General Resource Conditional Access record defines the conditional access to a general resource. There is one record per general resource/access combination.

Table 257. General Resource Conditional Access Record				
Field Name	Type	Position		Comments
		Start	End	
GRCACC_RECORD_TYPE	Int	1	4	Record type of the General Resources Conditional Access record (0507).
GRCACC_NAME	Char	6	251	General resource name as taken from the profile name.
GRCACC_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRCACC_CATYPE	Char	262	269	The type of conditional access checking that is being performed. Valid values are CONSOLE, TERMINAL, JESINPUT, SYSID, APPCPORT, SERVAUTH, PROGRAM, and CRITERIA.

Table 257. General Resource Conditional Access Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
GRCACC_CANAME	Char	271	278	The name of a conditional access element which is permitted access.
GRCACC_AUTH_ID	Char	280	287	The user ID or group name which has authority to the general resource.
GRCACC_ACCESS	Char	289	296	The authority of the conditional access element/user combination. Valid values are NONE, READ, UPDATE, CONTROL, and ALTER.
GRCACC_ACCESS_CNT	Int	298	302	The number of times that the general resource was accessed.
GRCACC_NET_ID	Char	304	311	The network name when GRCACC_CATYPE is APPCPORT.
GRCACC_CACRITERIA	Char	313	556	Access criteria or SERVAUTH IP data.

General resource filter data record (0508)

The General Resource Filter Data record defines the information used to create the filter described by this DIGTNMAP profile and identifies the associated user ID or criteria (DIGTCRIT) profile.

Table 258. General Resource Filter Data Record				
Field Name	Type	Position		Comments
		Start	End	
GRFLTR_RECORD_TYPE	Int	1	4	Record Type of the Filter Data record (0508).
GRFLTR_NAME	Char	6	251	General resource name as taken from the profile name.
GRFLTR_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRFLTR_LABEL	Char	262	293	The label associated with this filter.
GRFLTR_STATUS	Char	295	302	The status of this filter (TRUST) for filters that are trusted.
GRFLTR_USER	Char	304	549	The user ID or criteria profile name associated with this filter.
GRFLTR_CREATE_NAME	Char	551	1061	The issuer's or subject's name, or both, used to create this profile. For issuer's name only, the value has a trailing cent sign (¢); for subject's name only, the value has a preceding cent sign; for both, the value has a cent sign between the issuer's name and subject's name.

General resource distributed identity mapping data record (0509)

The General Resource Distributed Identity Mapping Data record defines the information used to create the mapping described by this IDIDMAP class profile and identifies the associated user ID.

Table 259. General Resource Distributed Identity Mapping Record				
Field Name	Type	Position		Comments
		Start	End	
GRDMAP_RECORD_TYPE	Int	1	4	Record Type of the General Resource Distributed Identity Mapping Data record (0509).
GRDMAP_NAME	Char	6	251	General resource name as taken from the profile name. Note: This value is stored in the RACF database in UTF-8 format. If possible, database unload changes this value to the EBCDIC format. If not possible, hexadecimal values are produced.
GRDMAP_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.

Table 259. General Resource Distributed Identity Mapping Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
GRDMAP_LABEL	Char	262	293	The label associated with this mapping.
GRDMAP_USER	Char	295	302	The RACF user ID associated with this mapping.
GRDMAP_DIDREG	Char	304	558	The registry name value associated with this mapping. Note: This value is stored in the RACF database in UTF-8 format. If possible, database unload changes this value to the EBCDIC format. If not possible, hexadecimal values are produced.

General resource session data record (0510)

The General Resource Session Data record defines the session data associated with a general resource. There is one record per APPCLU profile.

Table 260. General Resource Session Data Record				
Field Name	Type	Position		Comments
		Start	End	
GRSES_RECORD_TYPE	Int	1	4	Record type of the General Resources Session Data record (0510).
GRSES_NAME	Char	6	251	General resource name as taken from the profile name.
GRSES_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely APPCLU.
GRSES_SESSION_KEY	Char	262	269	The key associated with the APPC session.
GRSES_LOCKED	Char	271	274	Is the profile locked? Valid Values include "Yes" and "No".
GRSES_KEY_DATE	Date	276	285	Last date that the session key was changed.
GRSES_KEY_INTERVAL	Int	287	291	Number of days that the key is valid.
GRSES_SLS_FAIL	Int	293	297	Current number of failed attempts.
GRSES_MAX_FAIL	Int	299	303	Number of failed attempts before logout.
GRSES_CONVSEC	Char	305	312	Specifies the security checking performed when sessions are established. Valid values are NONE, CONVSEC, PERSISTV, ALREADYV, and AVPV.

General resource session entities record (0511)

The General Resource Session Entities record defines the entities associated with a general resource APPCLU profile. There is one record per APPCLU profile/session entity combination.

Table 261. General Resource Session Entity Record				
Field Name	Type	Position		Comments
		Start	End	
GRSESE_RECORD_TYPE	Int	1	4	Record type of the General Resources Session Entities record (0511).
GRSESE_NAME	Char	6	251	General resource name as taken from the profile name.
GRSESE_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely APPCLU.
GRSESE_ENTITY_NAME	Char	262	296	Entity name.
GRSESE_FAIL_CNT	Int	298	302	The number of failed session attempts.

General resource DLF data record (0520)

The General Resource DLF Data record defines the Data Lookaside Facility (DLF) data associated with a general resource. There is one record per general resource/DLF data combination.

Table 262. General Resource DLF Data Record				
Field Name	Type	Position		Comments
		Start	End	
GRDLF_RECORD_TYPE	Int	1	4	Record type of the General Resources DLF Data record (0520).
GRDLF_NAME	Char	6	251	General resource name as taken from the profile name.
GRDLF_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely DLFCLASS.
GRDLF_RETAIN	Char	262	265	Is this a retained resource? Valid Values include "Yes" and "No".

General resource DLF job names record (0521)

The General Resource DLF Job Names record defines the job names associated with a DLF general resource. There is one record per general resource/DLF job name combination.

Table 263. General Resource DLF Job Names Record				
Field Name	Type	Position		Comments
		Start	End	
GRDLFJ_RECORD_TYPE	Int	1	4	Record type of the General Resources DLF Job Names record (0521).
GRDLFJ_NAME	Char	6	251	General resource name as taken from the profile name.
GRDLFJ_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely DLFCLASS.
GRDLFJ_JOB_NAME	Char	262	269	The job name associated with the general resource.

General resource SSIGNON data record (0530)

The General Resource SSIGNON Data Record defines the method of protection for the encryption key of a general resource.

Table 264. General Resource SSIGNON Data Record				
Field Name	Type	Position		Comments
		Start	End	
GRSIGN_RECORD_TYPE	Int	1	4	Record type of the SSIGNON data record (0530).
GRSIGN_NAME	Char	6	251	General resource name as taken from the profile name.
GRSIGN_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRSIGN_PROTECTION	Char	262	325	<p>legacy PassTicket encryption key. Can contain one of the following values:</p> <ul style="list-style-type: none"> *MASKED* - KEYMASKED keyword was used *KEYTOKEN* - KEYENCRYPTED was used, and the key exists within a key token, perhaps due to an error with ICSF Key label name – KEYLABEL or KEYENCRYPTED was used, and the key is stored in ICSF with a key label. The output is the label name, which is a 64-character value padded with blanks if necessary. *UNKNOWN* - the format of the data is unrecognized.

Table 264. General Resource SSIGNON Data Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
GRSIGN_KEY_LABEL	Char	327	390	The enhanced PassTicket ICSF CKDS Key Label name.
GRSIGN_TYPE	Char	392	403	Enhanced PassTicket type.
GRSIGN_TIMEOUT	Int	405	414	Enhanced PassTicket timeout setting.
GRSIGN_REPLAY	Char	416	419	Indicates whether enhanced PassTicket replays are allowed.

General resource started task data record (0540)

The General Resource Started Task Data Record defines the information associated with the definition of a started task in the STARTED general resource class.

Table 265. General Resource Started Task Data Record				
Field Name	Type	Position		Comments
		Start	End	
GRST_RECORD_TYPE	Int	1	4	Record type (0540).
GRST_NAME	Char	6	251	Profile name.
GRST_CLASS_NAME	Char	253	260	The class name, STARTED.
GRST_USER_ID	Char	262	269	User ID assigned.
GRST_GROUP_ID	Char	271	278	Group name assigned.
GRST_TRUSTED	Char	280	283	Is process to run trusted? Valid Values include "Yes" and "No".
GRST_PRIVILEGED	Char	285	288	Is process to run privileged? Valid Values include "Yes" and "No".
GRST_TRACE	Char	290	293	Is entry to be traced? Valid Values include "Yes" and "No".

General resource SystemView data record (0550)

The General Resource SystemView Data Record defines the information associated with the SYSMVIEW general resource class.

Table 266. General Resource SystemView Data Record				
Field Name	Type	Position		Comments
		Start	End	
GRSV_RECORD_TYPE	Int	1	4	Record type (0550).
GRSV_NAME	Char	6	251	Profile name.
GRSV_CLASS_NAME	Char	253	260	Class name, SYSMVIEW.
GRSV_SCRIPT_NAME	Char	262	269	Logon script name for the application.
GRSV_PARM_NAME	Char	271	278	Parameter list name for the application.

General resource certificate data record (0560)

The general resource certificate data record defines the information associated with the digital certificate.

Table 267. General Resource Certificate Data Record				
Field Name	Type	Position		Comments
		Start	End	
GRCERT_RECORD_TYPE	Int	1	4	Record type of the Certificate Data record (0560).

Table 267. General Resource Certificate Data Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
GRCERT_NAME	Char	6	251	General resource name as taken from the profile name.
GRCERT_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRCERT_START_DATE	Date	262	271	The date from which this certificate is valid.
GRCERT_START_TIME	Time	273	280	The time from which this certificate is valid.
GRCERT_END_DATE	Date	282	291	The date after which this certificate is no longer valid.
GRCERT_END_TIME	Time	293	300	The time after which this certificate is no longer valid.
GRCERT_KEY_TYPE	Char	302	309	The type of key associated with the certificate. Valid values: BPECC, BPECCTKN, BPECTKNT, DSA, ICSFTOKN, NTECC, NTECCTKN, NTECTKNT, PCICCTKN, PKCSDER, PUBTOKEN, RSATKNT, or all blanks indicating no private key. The value PUBTOKEN indicates that the public key (without the private key) is stored in ICSF.
GRCERT_KEY_SIZE	Int	311	320	The size of private key associated with the certificate, expressed in bits.
GRCERT_LAST_SERIAL	Char	322	337	The hexadecimal representation of the low-order eight bytes of the serial number of the last certificate signed with this key.
GRCERT_RING_SEQN	Int	339	348	A sequence number for certificates within the ring.
GRCERT_GEN_REQ	Char	350	353	Indicator to show if the certificate is used to generate a request. Valid Values include "Yes" and "No".

General resource certificate references record (0561)

The general resource certificate references record identifies the key ring associated with the digital certificate.

Table 268. General Resource Certificate References Record				
Field Name	Type	Position		Comments
		Start	End	
CERTR_RECORD_TYPE	Int	1	4	Record type of the Certificate References record (0561).
CERTR_NAME	Char	6	251	General resource name as taken from the profile name.
CERTR_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
CERTR_RING_NAME	Char	262	507	The name of the profile which represents a key ring with which this certificate is associated.

General resource key ring data record (0562)

The general resource key ring data record defines the information associated with the key ring.

Table 269. General Resource Key Ring Data Record				
Field Name	Type	Position		Comments
		Start	End	
KEYR_RECORD_TYPE	Int	1	4	Record type of the Key Ring Data record (0562).
KEYR_NAME	Char	6	251	General resource name as taken from the profile name.
KEYR_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.

Table 269. General Resource Key Ring Data Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
KEYR_CERT_NAME	Char	262	507	The name of the profile which contains the certificate which is in this key ring.
KEYR_CERT_USAGE	Char	509	516	The usage of the certificate within the ring. Valid values are PERSONAL, SITE, and CERTAUTH.
KEYR_CERT_DEFAULT	Char	518	521	Is this certificate the default certificate within the ring? Valid Values include "Yes" and "No".
KEYR_CERT_LABEL	Char	523	554	The label associated with the certificate.

General resource TME data record (0570)

The General Resource TME data record identifies the parent ROLE profile from which this profile inherits attributes. There is one record per general resource profile/TME data combination.

Table 270. General Resource TME Data Record				
Field Name	Type	Position		Comments
		Start	End	
GRTME_RECORD_TYPE	Int	1	4	Record type of the general resource TME data record (0570).
GRTME_NAME	Char	6	251	General resource name as taken from the profile name.
GRTME_CLASS_NAME	Char	253	260	Name of the class to which the general resource belongs.
GRTME_PARENT	Char	262	507	Parent role.

General resource TME child record (0571)

The general resource TME child record identifies a ROLE profile which inherits attributes from this profile. There is one record per general resource/child combination.

Table 271. General Resource TME Child Record				
Field Name	Type	Position		Comments
		Start	End	
GRTMEC_RECORD_TYPE	Int	1	4	Record type of the general resource TME child record (0571).
GRTMEC_NAME	Char	6	251	General resource name as taken from the profile name.
GRTMEC_CLASS_NAME	Char	253	260	Name of the class to which the general resource belongs.
GRTMEC_CHILD	Char	262	507	Child role.

General resource TME resource record (0572)

The general resource TME resource record identifies resources and access authorities for groups defined in the role. There is one record per general resource/resource combination.

Table 272. General Resource TME Resource Record				
Field Name	Type	Position		Comments
		Start	End	
GRTMER_RECORD_TYPE	Int	1	4	Record type of the general resource TME resource record (0572).
GRTMER_NAME	Char	6	251	General resource name as taken from the profile name.
GRTMER_CLASS_NAME	Char	253	260	Name of the class to which the general resource belongs.
GRTMER_ORIGIN_ROLE	Char	262	507	Role profile from which resource access is inherited.

Table 272. General Resource TME Resource Record (continued)

Field Name	Type	Position		Comments
		Start	End	
GRTMER_PROF_CLASS	Char	509	516	Class name of the origin-role resource.
GRTMER_PROF_NAME	Char	518	763	Resource name defined in the origin role.
GRTMER_ACCESS_AUTH	Char	765	772	Access permission to the resource.
GRTMER_COND_CLASS	Char	774	781	Class name for conditional access.
GRTMER_COND_PROF	Char	783	1028	Resource profile for conditional access.

General resource TME group record (0573)

The general resource TME group record identifies groups that are permitted to resources in the role. There is one record per general resource/group combination.

Table 273. General Resource TME Group Record

Field Name	Type	Position		Comments
		Start	End	
GRTMEG_RECORD_TYPE	Int	1	4	Record type of the general resource TME group record (0573).
GRTMEG_NAME	Char	6	251	General resource name as taken from the profile name.
GRTMEG_CLASS_NAME	Char	253	260	Name of the class to which the general resource belongs.
GRTMEG_GROUP	Char	262	269	Group name defined to the role.

General resource TME role record (0574)

The general resource TME role record identifies ROLE profiles and access authorities referencing the general resource. There is one record per general resource/role combination.

Table 274. General Resource TME Role Record

Field Name	Type	Position		Comments
		Start	End	
GRTMEE_RECORD_TYPE	Int	1	4	Record type of the general resource TME role record (0574).
GRTMEE_NAME	Char	6	251	General resource name as taken from the profile name.
GRTMEE_CLASS_NAME	Char	253	260	Name of the class to which the general resource belongs.
GRTMEE_ROLE_NAME	Char	262	507	Role profile name.
GRTMEE_ACCESS_AUTH	Char	509	516	Access permission to this resource as defined by the role.
GRTMEE_COND_CLASS	Char	518	525	Class name for conditional access.
GRTMEE_COND_PROF	Char	527	772	Resource profile for conditional access.

General resource KERB data record (0580)

The general resource KERB Data record defines the Kerberos information for a realm. There is only one record per general resource profile that contains a KERB segment.

Table 275. General Resource KERB Data Record

Field Name	Type	Position		Comments
		Start	End	
GRKERB_RECORD_TYPE	Int	1	4	Record type of the general resource KERB segment record (0580).

Table 275. General Resource KERB Data Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
GRKERB_NAME	Char	6	251	General resource name as taken from the profile name.
GRKERB_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRKERB_KERBNAME	Char	262	501	The Kerberos realm name.
GRKERB_MIN_LIFE	Int	503	512	Minimum ticket life.
GRKERB_MAX_LIFE	Int	514	523	Maximum ticket life.
GRKERB_DEF_LIFE	Int	525	534	Default ticket life.
GRKERB_KEY_VERS	Int	536	538	Current key version.
GRKERB_ENCRYPT_DES	Char	540	543	Is key encryption using DES enabled? Valid Values include "Yes" and "No".
GRKERB_ENCRYPT_DES3	Char	545	548	Is key encryption using DES3 enabled? Valid Values include "Yes" and "No".
GRKERB_ENCRYPT_DESD	Char	550	553	Is key encryption using DES with derivation enabled? Valid Values include "Yes" and "No".
GRKERB_ENCRPT_A128	Char	555	558	Is key encryption using AES128 enabled? Valid Values include "Yes" and "No".
GRKERB_ENCRPT_A256	Char	560	563	Is key encryption using AES256 enabled? Valid Values include "Yes" and "No".
GRKERB_ENCRPT_A128SHA2	Char	565	568	Is key encryption using AES128 SHA2 enabled? Valid Values include "Yes" and "No".
GRKERB_ENCRPT_A256SHA2	Char	570	573	Is key encryption using AES256 SHA2 enabled? Valid Values include "Yes" and "No".
GRKERB_CHKADDRS	Char	620	623	Should the Kerberos server check addresses in tickets? Valid Values include "Yes" and "No".

General resource PROXY record (0590)

The general resource PROXY record identifies default information related to the LDAP proxy for a general resource. There is only one record per general resource profile that contains a PROXY segment.

Table 276. General Resource PROXY Record				
Field Name	Type	Position		Comments
		Start	End	
GRPROXY_RECORD_TYPE	Int	1	4	Record type of the general resource PROXY record (0590).
GRPROXY_NAME	Char	6	251	General resource name as taken from the profile name.
GRPROXY_CLASS_NAME	Char	253	260	Name of the class to which the general resource belongs.
GRPROXY_LDAP_HOST	Char	262	1284	LDAP server URL.
GRPROXY_BIND_DN	Char	1286	2308	LDAP BIND distinguished name.

General resource EIM record (05A0)

The general resource EIM record defines EIM-related information. There is only one record per general resource profile that contains an EIM segment.

Table 277. General Resource EIM Record

Field Name	Type	Position		Comments
		Start	End	
GREIM_RECORD_TYPE	Int	1	4	Record type of the general resource EIM segment record (05A0).
GREIM_NAME	Char	6	251	Profile name.
GREIM_CLASS_NAME	Char	253	260	Class name.
GREIM_DOMAIN_DN	Char	262	1284	EIM domain name.
GREIM_ENABLE	Char	1286	1289	EIM Enable option. Valid Values include "Yes" and "No".
	Char	1291	1364	Reserved for IBM's use.
GREIM_LOCAL_REG	Char	1366	1620	EIM LDAP local registry name.
GREIM_KERBREG	Char	1622	1876	EIM Kerberos Registry Name
GREIM_X509REG	Char	1878	2132	EIM X.509 Registry name

General resource alias data record (05B0)

Table 278. General Resource Alias Data Record

Field Name	Type	Position		Comments
		Start	End	
GRALIAS_RECORD_TYPE	Int	1	4	Record type of the general resource ALIAS group record (05B0).
GRALIAS_NAME	Char	6	251	General resource name as taken from the profile.
GRALIAS_CLASS_NAME	Char	253	260	Name of the class to which the general resource belongs.
GRALIAS_IPLOOK	Int	262	293	IP lookup value in SERVAUTH class.

General resource CDTINFO data record (05C0)

The general resource CDTINFO data record defines class descriptor table information. There is only one record per general resource profile that contains a CDTINFO segment.

Table 279. General Resource CDTINFO Data Record

Field Name	Type	Position		Comments
		Start	End	
GRCDT_RECORD_TYPE	Int	1	4	Record type of the general resource CDTINFO data record (05C0).
GRCDT_NAME	Char	6	251	General resource name as taken from the profile.
GRCDT_CLASS_NAME	Char	253	260	Name of the class to which the general resource belongs, namely CDT.
GRCDT_POSIT	Int	262	271	POSIT number for class.
GRCDT_MAXLENGTH	Int	273	275	Maximum length of profile names when using ENTITYX.
GRCDT_MAXLENX	Int	277	286	Maximum length of profile names when using ENTITYX.
GRCDT_DEFAULTRC	Int	288	290	Default return code.
GRCDT_KEYQUALIFIER	Int	292	301	Number of key qualifiers.
GRCDT_GROUP	Char	303	310	Resource grouping class name.
GRCDT_MEMBER	Char	312	319	Member class name.
GRCDT_FIRST_ALPHA	Char	321	324	Is an alphabetic character allowed in the first character of a profile name? Valid Values include "Yes" and "No".

Table 279. General Resource CDTINFO Data Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
GRCDT_FIRST_NATL	Char	326	329	Is a national character allowed in the first character of a profile name? Valid Values include "Yes" and "No".
GRCDT_FIRST_NUM	Char	331	334	Is a numeric character allowed in the first character of a profile name? Valid Values include "Yes" and "No".
GRCDT_FIRST_SPEC	Char	336	339	Is a special character allowed in the first character of a profile name? Valid Values include "Yes" and "No".
GRCDT_OTHER_ALPHA	Char	341	344	Is an alphabetic character allowed in other characters of a profile name? Valid Values include "Yes" and "No".
GRCDT_OTHER_NATL	Char	346	349	Is a national character allowed in other characters of a profile name? Valid Values include "Yes" and "No".
GRCDT_OTHER_NUM	Char	351	354	Is a numeric character allowed in other characters of a profile name? Valid Values include "Yes" and "No".
GRCDT_OTHER_SPEC	Char	356	359	Is a special character allowed in other characters of a profile name? Valid Values include "Yes" and "No".
GRCDT_OPER	Char	361	364	Is OPERATIONS attribute to be considered? Valid Values include "Yes" and "No".
GRCDT_DEFAULTUACC	Char	366	373	Default universal access. Valid values are ACEE, ALTER, CONTROL, UPDATE, READ, EXECUTE, NONE.
GRCDT_RACLIST	Char	375	384	RACLIST setting. Valid values are ALLOWED, DISALLOWED, REQUIRED.
GRCDT_GENLIST	Char	386	395	GENLIST setting. Valid values are ALLOWED, DISALLOWED.
GRCDT_PROF_ALLOW	Char	397	400	Are profiles allowed in the class? Valid Values include "Yes" and "No".
GRCDT_SECL_REQ	Char	402	405	Are security labels required for the class when MLACTIVE is on? Valid Values include "Yes" and "No".
GRCDT_MACPROCESS	Char	407	414	Type of mandatory access check processing. Valid values are EQUAL, NORMAL, REVERSE.
GRCDT_SIGNAL	Char	416	419	Is ENF signal to be sent? Valid Values include "Yes" and "No".
GRCDT_CASE	Char	421	428	Case of profile names. Valid values are ASIS, UPPER.
GRCDT_GENERIC	Char	430	439	GENERIC setting. Valid values are ALLOWED and DISALLOWED.

General resource ICTX data record (05D0)

The General Resource ICTX record contains the configuration options used to control the behavior of the ICTX identity cache.

Table 280. General Resource ICTX Data Record				
Field Name	Type	Position		Comments
		Start	End	
GRICTX_RECORD_TYPE	Int	1	4	Record type of the general resource ICTX segment record (05D0).
GRICTX_NAME	Char	6	251	General resource name as taken from the profile name.
GRICTX_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRICTX_USEMAP	Char	262	265	Should the identity cache store an application provided identity mapping? Valid Values include "Yes" and "No".

Table 280. General Resource ICTX Data Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
GRICTX_DOMAP	Char	267	270	Should the identity cache determine and store the identity mapping? Valid Values include "Yes" and "No".
GRICTX_MAPREQ	Char	272	275	Is an identity mapping required? Valid Values include "Yes" and "No".
GRICTX_MAP_TIMEOUT	Int	277	281	How long the identity cache should store an identity mapping.

General Resource CFDEF Data record (05E0)

The General Resource CFDEF Data record (05E0) defines custom field information. There is one record per general resource profile that contains a CFDEF segment.

Table 281. General Resource CFDEF Data Record				
Field Name	Type	Position		Comments
		Start	End	
GRCFDEF_RECORD_TYPE	Int	1	4	Record type of the general resource CFDEF data record (05E0).
GRCFDEF_NAME	Char	6	251	General resource name as taken from the profile name.
GRCFDEF_CLASS	Char	253	260	Name of the class to which the general resource belongs, namely CFIELD.
GRCFDEF_TYPE	Char	262	265	Data type for the custom field. Valid values are CHAR, FLAG, HEX, NUM.
GRCFDEF_MAXLEN	Int	267	276	Maximum length of the custom field.
GRCFDEF_MAXVAL	Int	278	287	Maximum value of the custom field.
GRCFDEF_MINVAL	Int	289	298	Minimum value of the custom field.
GRCFDEF_FIRST	Char	300	307	Character restriction for the first character. Valid values are ALPHA, ALPHANUM, ANY, NONATABC, NONATNUM, NUMERIC.
GRCFDEF_OTHER	Char	309	316	Character restriction for other characters. Valid values are ALPHA, ALPHANUM, ANY, NONATABC, NONATNUM, NUMERIC.
GRCFDEF_MIXED	Char	318	321	Is mixed case allowed in the field? Valid values are "Yes" and "No".
GRCFDEF_HELP	Char	323	577	Help text for the custom field.
GRCFDEF_LISTHEAD	Char	579	618	List heading for the custom field.
GRCFDEF_VALREXX	Char	620	627	Name of the REXX exec to validate the custom field value.
GRCFDEF_ACEE	Char	629	632	For USER profile fields, is this field to be made available in an ACEE created for the user? Valid values are "Yes" and "No".

General Resource SIGVER data record (05F0)

The General Resource SIGVER Data record (05F0) defines the settings that control program signature verification. There is one record per general resource profile that contains a SIGVER segment.

Table 282. General Resource SIGVER Data Record				
Field Name	Type	Position		Comments
		Start	End	
GRSIG_RECORD_TYPE	Int	1	4	Record type of the general resource SIGVER data record (05F0).

Table 282. General Resource SIGVER Data Record (continued)				
Field Name	Type	Position		Comments
		Start	End	
GRSIG_NAME	Char	6	251	General resource name as taken from the profile name.
GRSIG_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRSIG_SIGREQUIRED	Char	262	265	Signature required. Valid Values include "Yes" and "No".
GRSIG_FAILLOAD	Char	267	276	Condition for which load should fail. Valid values are NEVER, BADSIGONLY, and ANYBAD.
GRSIG_AUDIT	Char	278	287	Condition for which RACF should audit. Valid values are NONE, ALL, SUCCESS, BADSIGONLY, and ANYBAD.

General Resource ICSF record (05G0)

The General Resource ICSF record (05G0) defines the ICSF attributes associated with a general resource profile. There is one record per general resource/ICSF data combination.

Table 283. General Resource ICSF Record				
Field Name	Type	Position		Comments
		Start	End	
GRCSF_RECORD_TYPE	Int	1	4	Record type of the general resource ICSF record (05G0).
GRCSF_NAME	Char	6	251	General resource name as taken from the profile name.
GRCSF_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRCSF_EXPORTABLE	Char	262	273	Is the symmetric key exportable? Valid values are: BYNONE, BYLIST, and BYANY.
GRCSF_USAGE	Char	275	529	Allowable uses of the asymmetric key. Valid values are: HANDSHAKE, NOHANDSHAKE, SECUREEXPORT, and NOSECUREEXPORT.
GRCSF_CPACF_WRAP	Char	531	533	Specifies whether the encrypted symmetric key is eligible to be rewrapped by CP Assist for Cryptographic Function (CPACF). Valid Values include "Yes" and "No".
GRCSF_CPACF_RET	Char	535	537	Specifies whether the encrypted symmetric keys that are rewrapped by CP Assist for Cryptographic Function (CPACF) are eligible to be returned to an authorized caller.

General Resource ICSF key label record (05G1)

The General Resource ICSF key label record (05G1) defines the PKDS key labels associated with an ICSF general resource. There is one record per general resource/ICSF key label combination.

Table 284. General Resource ICSF key label Record				
Field Name	Type	Position		Comments
		Start	End	
GRCSFK_RECORD_TYPE	Int	1	4	Record type of the general resource ICSF key label record (05G1).
GRCSFK_NAME	Char	6	251	General resource name as taken from the profile name.
GRCSFK_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRCSFK_LABEL	Char	262	325	ICSF key label of a public key that can be used to export this symmetric key.

General Resource ICSF certificate identifier record (05G2)

The General Resource ICSF certificate identifier record (05G2) defines the certificates associated with an ICSF general resource. There is one record per general resource/certificate combination.

Table 285. General Resource ICSF certificate identifier Record				
Field Name	Type	Position		Comments
		Start	End	
GRCSFC_RECORD_TYPE	Int	1	4	Record type of the general resource ICSF certificate identifier record (05G2).
GRCSFC_NAME	Char	6	251	General resource name as taken from the profile name.
GRCSFC_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRCSFC_LABEL	Char	262	358	Certificate identifier of a public key that can be used to export this symmetric key.

General resource MFA factor definition record (05H0)

The General resource MFA factor definition record defines the basic information about a the MFA factor definition record.

Table 286. General resource MFA factor definition record				
Field Name	Type	Position		Comments
		Start	End	
GRMFA_RECORD_TYPE	Int	1	4	Record type of the Multifactor factor definition data record (05H0)
GRMFA_NAME	Char	6	251	General resource name as taken from the profile name.
GRMFA_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely MFADEF.
GRMFA_FACTOR_DATA_LEN	Int	262	266	Length of factor data.

General resource MFPOLICY definition record (05I0)

General resource MFPOLICY definition record (05I0)

Table 287. General resource MFAPOLICY definition record (05I0)				
Field Name	Type	Position		Comments
		Start	End	
GRMFP_RECORD_TYPE	Int	1	4	Record type of the Multifactor Policy Definition data record (05I0).
GRMFP_NAME	Char	6	251	General resource name as taken from the profile name.
GRMFP_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely MFADEF.
GRMFP_TOKEN_TIMEOUT	Int	262	271	MFA token timeout setting.
GRMFP_REUSE	Yes/No	273	275	MFA token reuse setting.

General resource MFA policy factors record (05I1)

The general resource MFA policy factors record (05I1)

Table 288. General resource MFA policy factors record (05I1)				
Field Name	Type	Position		Comments
		Start	End	
GRMPF_RECORD_TYPE	Int	1	4	Record type of the user Multifactor authentication policy factors record (05I1).
GRMPF_NAME	Char	6	251	General resource name as taken from the profile name.
GRMPF_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely MFADEF.
GRMPF_POL_FACTOR	Char	262	281	Policy factor name.

General resource CSDATA record (05J1)

Table 289. General resource CSDATA record (05J1)				
Field Name	Type	Position		Comments
		Start	End	
GRCSO_RECORD_TYPE	Int	1	4	Record type of the General Resources CSDA custom fields record (05J1).
GRCSO_NAME	Char	6	251	General resource name as taken from the profile name.
GRCSO_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRCSO_TYPE	Char	262	265	Data type for the custom field. Valid values are CHAR, FLAG, HEX, NUM.
GRCSO_KEY	Char	267	298	Custom field keyword; maximum length = 8.
GRCSO_VALUE	Char	300	1399	Custom field value.

General resource IDTPARMS definition record (05k0)

Table 290. General resource IDTPARMS definition record (05k0)				
Field Name	Type	Position		Comments
		Start	End	
GRIDTP_RECORD_TYPE	Int	1	4	Record type of the Identity Token data record (05K0).
GRIDTP_NAME	Char	6	251	General resource name as taken from the profile name.
GRIDTP_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely IDTDATA.
GRIDTP_SIG_TOKEN_NAME	Char	262	293	The ICSF PKCS#11 token name.
GRIDTP_SIG_SEQ_NUM	Char	295	302	The ICSF PKCS#11 sequence number.
GRIDTP_SIG_CAT	Char	304	307	The ICSF PKCS#11 category.
GRIDTP_SIG_ALG	Char	309	340	The signature algorithm.
GRIDTP_TIMEOUT	Int	342	351	IDT timeout setting.
GRIDTP_ANYAPPL	Char	353	355	Is the IDT allowed for any application? Valid values include "Yes" and "No".
GRIDTP_PROTALLOWED	Char	358	361	Is the IDT allowed to authenticate a protected user? Valid values include "Yes" and "No".
GRIDTP_SIG_LABEL_PRI	Char	363	426	ICSF label name of the primary key.
GRIDTP_SIG_KID_PRI	Char	428	459	Key identifier of the primary key.

General resource JES data record (05L0)

The General Resource JES Data Record defines the JES segment information associated with a general resource.

Table 291. General Resource JES Data Record				
Field Name	Type	Position		Comments
		Start	End	
GRJES_RECORD_TYPE	Int	1	4	Record type of the JES data record (05L0).
GRJES_NAME	Char	6	251	General resource name as taken from the profile name.
GRJES_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRJES_KEYLABEL	Char	262	325	The label of the ICSF key that is used to encrypt the JES spool data.

General resource certificate information record (1560)

The general resource certificate information record (1560) defines additional information associated with the digital certificate.

Table 292. General resource certificate information record				
Field Name	Type	Position		Comments
		Start	End	
CERTN_RECORD_TYPE	Int	1	4	Record type of the general resource certificate information record (1560).
CERTN_NAME	Char	6	251	General resource name as taken from the profile name.
CERTN_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
CERTN_ISSUER_DN	Char	262	1285	Issuers distinguished name.
CERTN_SUBJECT_DN	Char	1287	2310	Subjects distinguished name.
CERTN_SIG_ALG	Char	2312	2327	Certificate signature algorithm. Valid values are md2RSA, md5RSA, sha1RSA, sha1DSA, sha256RSA, sha224RSA, sha384RSA, sha512RSA, sha1ECDSA, sha256ECDSA, sha224ECDSA, sha384ECDSA, sha512ECDSA, sha1RSAPSS, sha224RSAPSS, sha384RSAPSS, sha512RSAPSS, and UNKNOWN.
CERTN_CERT_FGRPRNT	Char	2329	2392	Certificate SHA256 fingerprint in printable hex.

Chapter 10. The RACF PassTicket

The RACF PassTicket is a *one-time-only* password that is generated by a requesting product or function. It is an alternative to the RACF password that removes the need to send RACF passwords across the network in clear text. It makes it possible to move the authentication of a mainframe application user ID from RACF to another authorized function executing on the host system or to the work station local area network (LAN) environment.

RACF provides support for the following PassTicket functions:

- Generating a PassTicket.
- Evaluating a PassTicket.

RACF PassTickets can be configured with two different algorithms:

- The legacy PassTicket algorithm
- The enhanced PassTicket algorithm

The legacy PassTicket algorithm is the original PassTicket implementation and uses a DES secret key. The enhanced PassTicket algorithm is an updated version of the PassTicket algorithm and uses an HMAC secret key. RACF supports generation and evaluation of PassTickets with either the legacy PassTicket algorithm or the enhanced PassTicket algorithm based on system configuration. IBM highly recommends using the enhanced PassTicket algorithm as it provides the same capabilities as the legacy PassTicket algorithm but also provides increased security.

For more information on configuring PassTickets, see “The RACF PassTicket” in the *z/OS Security Server RACF Security Administrator's Guide*

Generating and evaluating a PassTicket

A product or function that generates a PassTicket must use the RACF legacy PassTicket generator algorithm or enhanced PassTicket generation algorithm. These algorithms require specific information as input data and produces a PassTicket that substitutes for a specific end-user RACF password. RACF uses the PassTicket to authenticate the end-user for a specific application running on a specific system that uses RACF for identification and authentication.

There are four ways to generate and evaluate a PassTicket using the legacy PassTicket algorithm or enhanced PassTicket algorithm:

- If the function using PassTickets is running on a z/OS system, you can use the RACF PassTicket-generation service (RCVTPTGN) to generate the PassTicket. The algorithm is already incorporated into the service and allows RACF to generate a PassTicket on the host. An authorized program, such as one authorized by the authorized program facility (APF), can use the service to generate PassTickets. See [“Using the RCVTPTGN service to generate a PassTicket” on page 420](#) for more information.
- For any function that generates a PassTicket, you can create a program that incorporates the algorithm. See [“Incorporating the PassTicket generator algorithm into your program” on page 421](#) for more information.
- You can use the R_ticketerv and R_GenSec callable services. This interface supports problem state callers, and both 31-bit and 64-bit callers. For more information about these callable services, see [R_ticketerv \(IRRSPK00\): Parse or extract](#) and [R_GenSec \(IRRS00 or IRRSGS64\): Generic security API interface in z/OS Security Server RACF Callable Services](#).
- Java™ code can use a Java interface that uses a Java Native Interface (JNI) and calls the R_ticketerv and R_GenSec callable services. For information about this interface, see the JavaDoc shipped in the IRRRacDoc.jar file, which is installed into the directory /usr/include/java_classes. Download the jar file to a workstation, un-jar it, and read it with a Web browser.

Using the RCVTPTGN service to generate a PassTicket

To allow RACF to authenticate a user with a PassTicket instead of a password, the non-RACF function performing the authentication calls the RCVTPTGN service to build a PassTicket.

The RCVTPTGN service:

- Is branch-entered by callers.
- Is *not* supported in cross-memory mode. Access register (AR) mode must use address space control (ASC).
- Is not supported in SRB mode.
- Requires that the caller be in key zero.
- Is unable to generate PassTickets using the PTKTDATA profiles which are qualified by user id and / or group. It can only generate PassTickets using profiles which match the application name.
- Supports generation of legacy PassTickets or enhanced PassTickets based on RACF configuration.

Before calling the PassTicket-generation service, the application must locate the address of the service. You can find this address from field RCVTPTGN in the RACF communications vector table (RCVT). The ICHPRCVT macro maps the RCVT and field CVTRAC points to it in the MVS communications vector table (CVT).

How the PassTicket-generation service works

The service:

- Uses standard linkage
- Uses the current system time, expressed in Greenwich Mean Time (GMT), ¹ as input for the algorithm
- Returns the PassTicket in general purpose register 0 (the leftmost four characters) and general purpose register 1 (the rightmost four characters)
- The type of PassTicket returned is based on the keys configured in the associated PTKTDATA class profile:
 - An enhanced PassTicket is returned when an enhanced PassTicket key label is configured with the EPTKEYLABEL keyword.
 - A legacy PassTicket is returned when a legacy PassTicket key is configured with the KEYMASKED, KEYENCRYPTED or KEYLABEL keywords and no enhanced PassTicket key label is configured.
 - In the case where a PTKTDATA class profile is configured to contain both a legacy PassTicket key and enhanced PassTicket key, an enhanced PassTicket is returned.
- Provides return codes
 - If a PassTicket is produced, register 15 contains a return code of 0
 - If a PassTicket is not produced, register 15 contains return code of 8
 - Register 0 contains a reason code. The 1st byte of the reason code indicates the problem, the other 3 bytes may contain additional information:

Value (decimal)	Meaning	Bytes 2-4
12	ICSF CSNBENC service failed	Byte 2=ICSF RC Byte 3 and 4=ICSF RSN
16	RACROUTE REQUEST=EXTRACT, TYPE=ENCRYPT failed	Byte 2=SAFRC from RACROUTE Bytes 3 and 4=0

¹ GMT is also referred to as coordinated universal time (UTC).

Value (decimal)	Meaning	Bytes 2-4
20	PTKTDATA class inactive	0
24	No profiles defined to the PTKTDATA class	0
28	Unable to load ICSF CSFACEE or CSFIQF service	Byte 2=Reason code from z/OS LOAD macro
36	PTKTDATA profile representing the APPL not found or the PTKTDATA profile does not have a key saved in the SSIGNON segment	0
52	Caller not in key 0	0
56	ICSF not initialized	Byte 2=ICSF RC Byte 3 and 4=ICSF RSN
60	ICSF CSNBHMG service failed.	Byte 2=ICSF RC Byte 3 and 4=ICSF RSN
Other = Internal error		

Notes:

1. Register 13 must point to a standard save area.
2. No additional recovery processing is provided by the PassTicket-generation service beyond what is already in effect within the invoking program.

Invoking the PassTicket-generation service

Following is an example of a generalized programming technique you can use with assembler language to invoke a service. It is not intended to be syntactically correct.

```
L 15,RCVTPTGN
CALL (15),(userid,appname)
```

where:

userid

Is the RACF user ID of the user the PassTicket authenticates. This field is a maximum of 9 bytes. The first byte contains the length of the non-blank portion of the *userid* field that follows. Bytes 2 through 9 contain the user ID and must be in uppercase and left-justified in the field.

appname

Is the application name that the PassTicket-generation service uses to locate the key used in the PassTicket generator algorithm. This field is a maximum of 9 bytes. The first byte is the length of the non-blank portion of the *appname* field that follows. Bytes 2 through 9 contain the application name and must be in uppercase and left-justified in the field.

When the service is invoked, only the *appname* (not the *userid* or *group*) is used to locate the PassTicket key. It is not possible to use the RCVTPTGN service to generate PassTickets using keys which are stored in user id or group id qualified profiles.

Incorporating the PassTicket generator algorithm into your program

To generate a PassTicket without using the RACF service, callable services, or Java interface, you need to incorporate either the RACF legacy PassTicket generator algorithm or enhanced PassTicket generator algorithm into your program.

The RACF PassTicket algorithms each consist of two parts:

- The RACF PassTicket generator
- The RACF PassTicket time-coder

The time-coder is invoked from within the RACF PassTicket generator and returns its results to the generator.

The flowcharts in [Figure 6 on page 422](#), [Figure 7 on page 423](#), [Figure 8 on page 424](#), [Figure 9 on page 425](#) and the descriptions that follow show how to implement the RACF legacy and enhanced PassTicket generator algorithms.

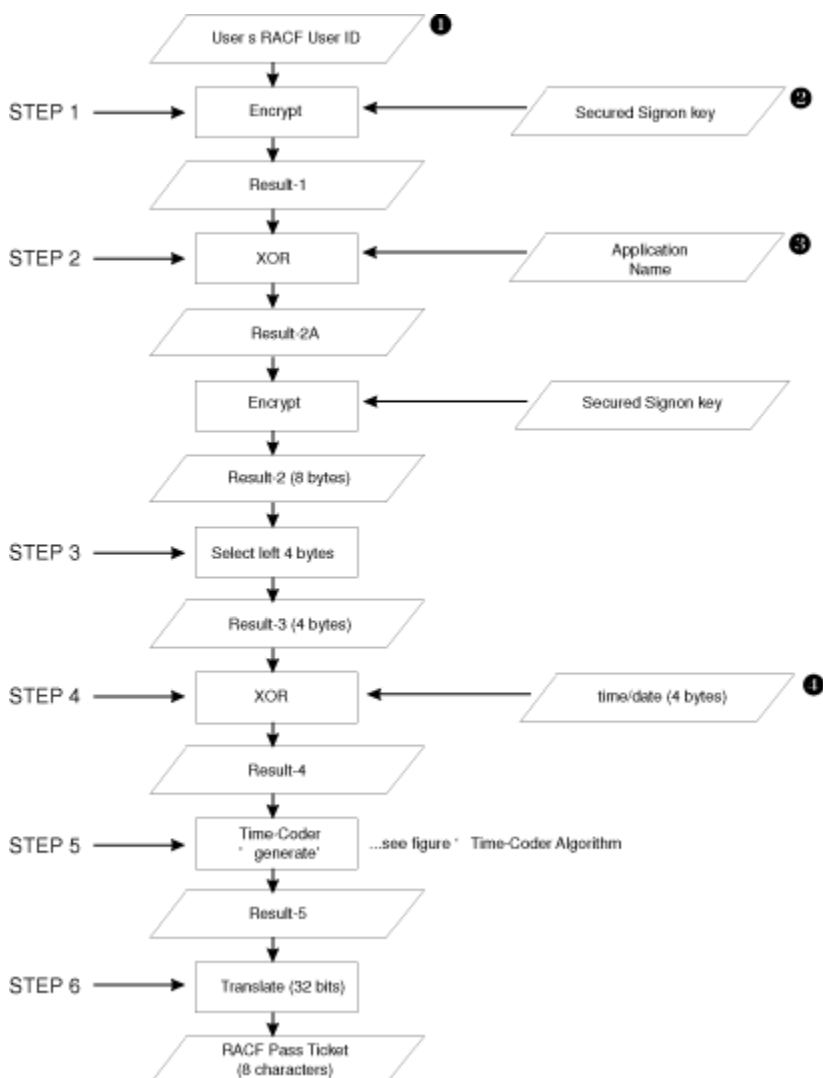


Figure 6. RACF legacy PassTicket generator for secured signon

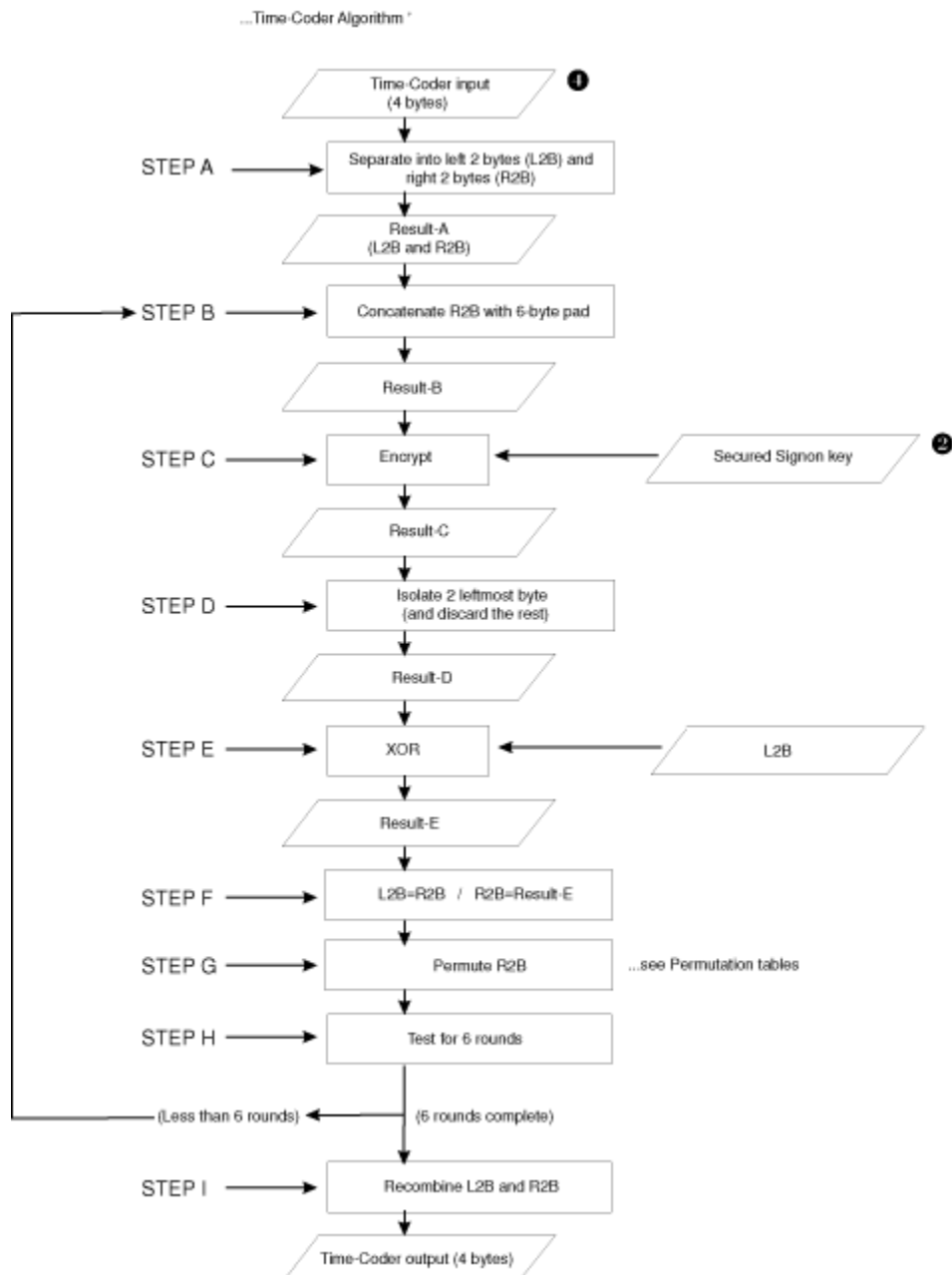


Figure 7. Algorithm for RACF legacy PassTicket time-coder used for secured signon

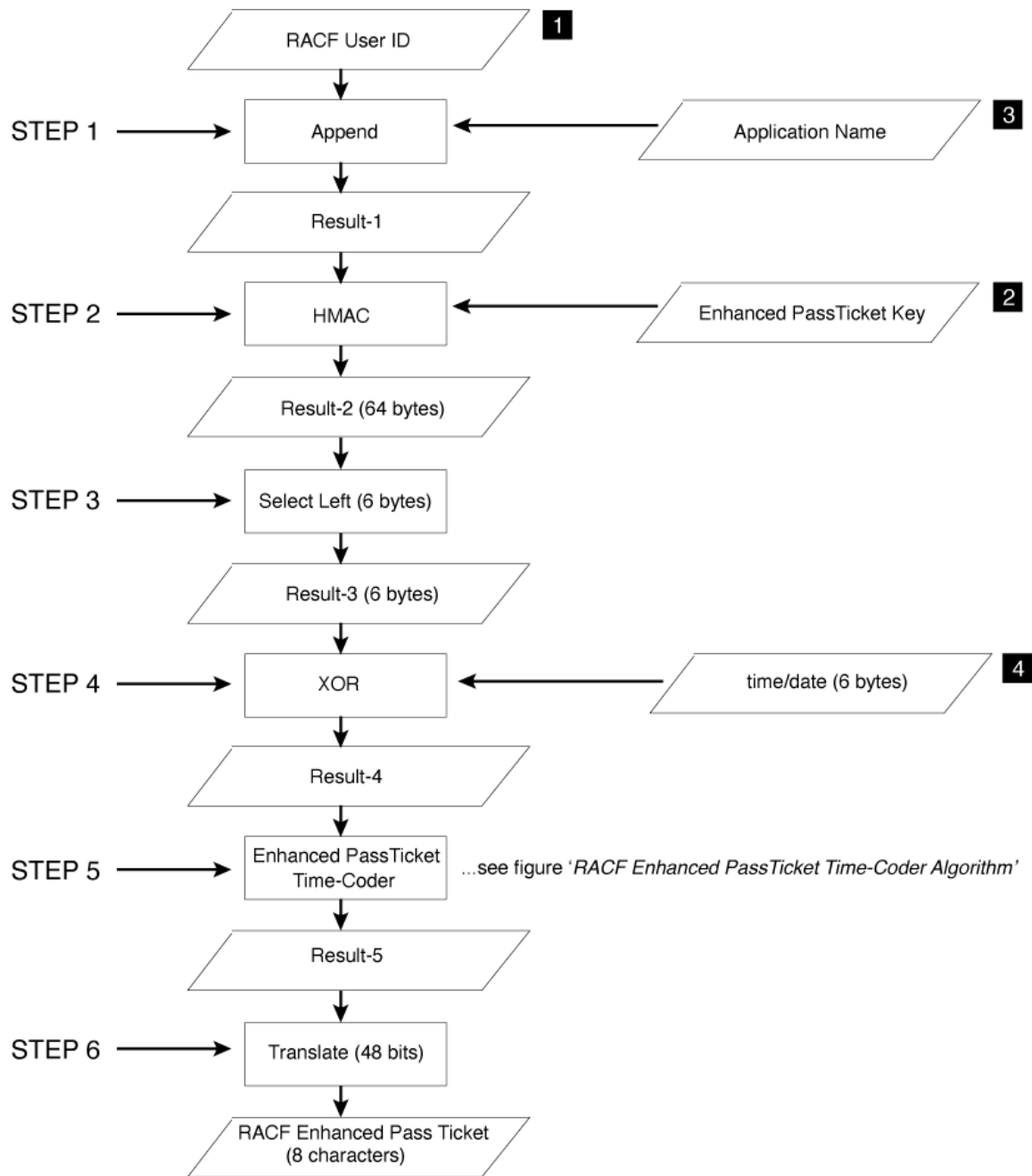


Figure 8. RACF enhanced PassTicket generator algorithm

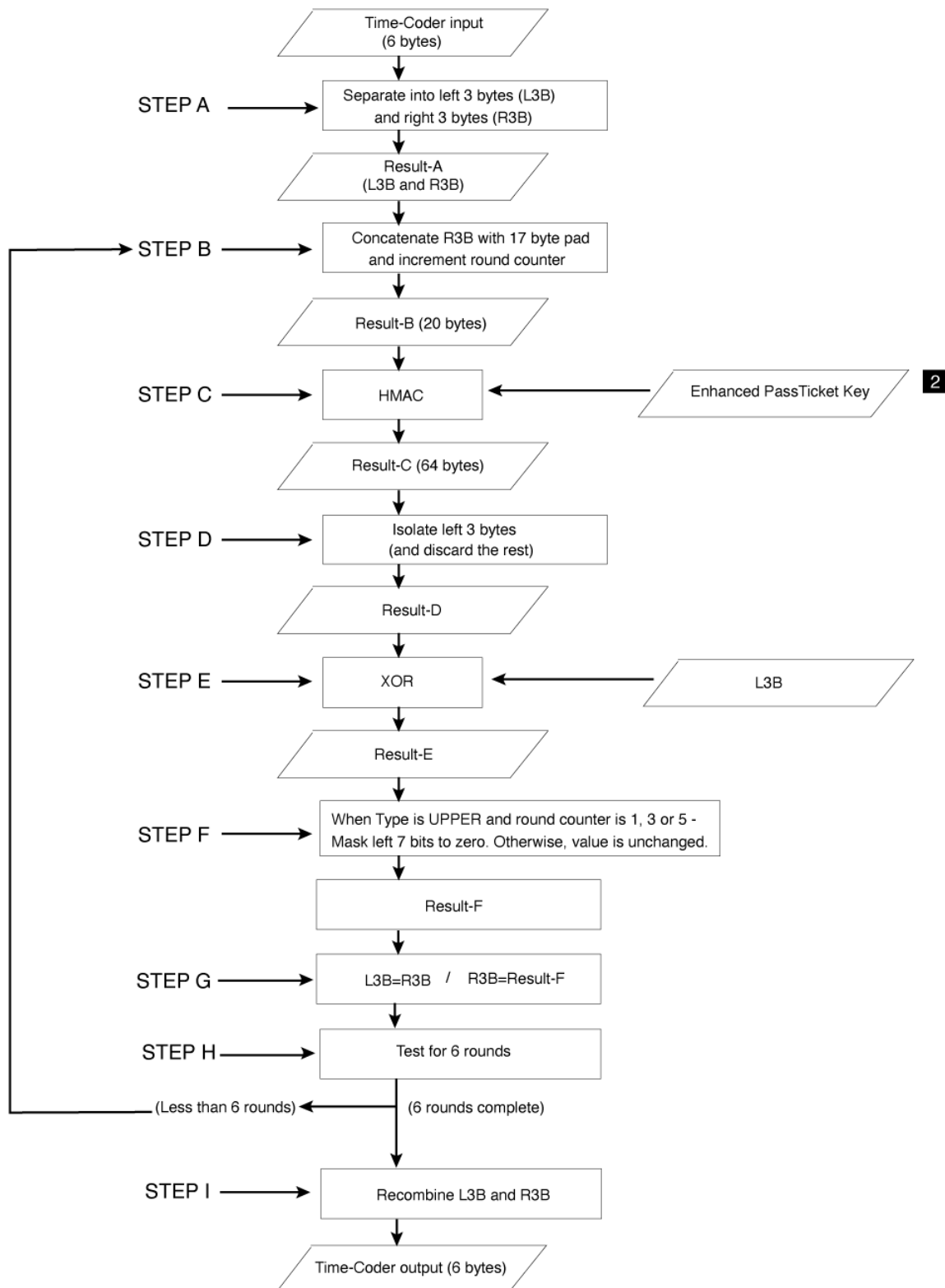


Figure 9. RACF enhanced PassTicket time-coder algorithm

Input data for the generator algorithms

To successfully use the PassTicket, the target application using RACF to identify and authenticate a user ID needs to have specific information for processing according to the algorithm. As shown in [Figure 7 on page 423](#), these are:

- A RACF host user ID

- The RACF PassTicket application key
- The application name
- Time and date information
- The PassTicket algorithm type

1

The RACF user ID:

- Identifies the user ID on the system on which the target application runs
- Is represented in EBCDIC
- Is left-justified and padded with blanks on the right to a length of 8 bytes

2

The RACF PassTicket application key:

- Must match the key value used when defining the application to the PTKTDATA class to RACF
- For the legacy PassTicket algorithm:
 - This is a DES secret key.
 - Contains only the characters 0 through 9 and A through F
- For the enhanced PassTicket algorithm:
 - This is an HMAC secret key.

3

The application name as defined for a particular application. You can use it to associate a PassTicket key with a particular host application. See *z/OS Security Server RACF Security Administrator's Guide* for information about determining application names.

The name:

- Is represented in EBCDIC
- Is left-justified and padded with blanks on the right to a length of 8 bytes

4

Time and date information:

This information:

- For the legacy PassTicket algorithm:
 - Must be a 4-byte binary number
- For the enhanced PassTicket algorithm:
 - Must be a 6-byte binary number
- Shows how many seconds elapsed since January 1, 1970, at 0000 Greenwich Mean Time (GMT)

Several programming languages support a function for representing time in this way. In C language, for example, you can obtain the time in this way:

1. Declare the variable *ts* as **long**.
2. Invoke the function **time(&ts)**.

This produces the number of seconds that elapsed since January 1, 1970 at 0000 GMT, expressed as an unsigned long integer.

Notes:

1. It is likely that the computer that authenticates the PassTicket is not the computer that generated it. To provide for differences in their internal clocks, the algorithms allow the generated time

to be different than the computer that is evaluating the PassTicket. For legacy PassTickets, the generated time must be within 10 minutes on either side of the TOD clock. For enhanced PassTickets, the amount of time skew is configurable in the PTKTDATA class profile.

2. For RACF to properly evaluate PassTickets, the TOD clock must be properly set to GMT rather than local time.

5

The PassTicket algorithm type:

- Identifies the type of algorithm used to generate and evaluate the PassTicket.
- The legacy PassTicket algorithm type is the original PassTicket algorithm and uses a DES secret key.
- The enhanced PassTicket algorithm type is an improved PassTicket algorithm and uses an HMAC secret key. An enhanced PassTicket can be generated with either a MIXED or UPPER character set.

How the legacy PassTicket generator algorithm works

The RACF legacy PassTicket generator algorithm uses the input information to create a legacy PassTicket. By using cryptographic techniques, the algorithm ensures that each PassTicket is unpredictable.

The legacy PassTicket is an 8-character alphanumeric string that can contain the characters A through Z and 0 through 9. The actual legacy PassTicket depends on the input values.

The following steps describe this process (see [Figure 6 on page 422](#) for a summary):

1. The RACF user ID **1** is encrypted using the RACF secured signon application key **2** as the encryption key to produce Result-1.

Note: All encryptions use the US National Institute of Standards and Technology Data Encryption Standard (DES) algorithm. Only the DES algorithm encoding is involved. You cannot perform general encryption and decryption of data with this implementation.

2. Result-1 from the first encryption is XORed ² with the application name **3**. The application name must be 8 bytes of EBCDIC characters with trailing blanks. The result (Result-2A) is encrypted using the application key value **2** as the encryption key to produce Result-2.

Note: If you understand cryptographic techniques, you should recognize the flow (Steps 1 and 2) to be a common cryptographic architecture (CCA) standard message authentication code algorithm.

3. The left 4 bytes from Result-2 of the second encryption are selected as input to the next step. The rest are discarded.
4. The resulting 4 bytes (Result-3) are XORed with the time and date information **4**. The time and date is in the form of a 4-byte field that contains the number of seconds that elapsed since January 1, 1970 at 0000 GMT in the form of a binary integer. (See [“Input data for the generator algorithms” on page 425](#) for a complete description.)
5. The result (Result-4) of that procedure is passed to the time-coder routine. Refer to the diagram in [Figure 7 on page 423](#) and to [“How the legacy PassTicket time-coder algorithm works” on page 427](#) to understand that process.
6. The result (Result-5) of the time-coder routine is translated, using a translation table described in [“The translation table” on page 429](#), to an 8-character string called the PassTicket. It is used in the user's host application signon request instead of the user's regular RACF password.

How the legacy PassTicket time-coder algorithm works

The RACF legacy PassTicket time-coder algorithm uses the result of Step [“4” on page 427](#) of the legacy generator algorithm. It creates the time-coder information and passes it back to step [“6” on page 427](#) of that algorithm.

² XOR is a Boolean function that processes two-bit strings of the same length, producing a third string of the same length as the output. In the output string, a bit is ON if the corresponding bit is ON for one of the input bit strings, but not both.

The following steps, which make up Step “5” on page 427 of the generator algorithm, shown in [Figure 7](#) on page 423 describe this process:

Step A

Separate the 4-byte time-coder input (Result-4) into two portions, L2B (the left side), and R2B (the right side) to produce Result-A.

Step B

Concatenate R2B (the right 2 bytes from Result-A) with 6 bytes of padding bits to form Result-B. In the resulting 8-byte string, the 2 bytes of R2B occupy the leftmost 2 byte positions.

The padding bits consist of two separate 6 byte strings: PAD1 and PAD2. PAD1 is the left half and PAD2 is the right half of a 12 byte string consisting of the user ID (from Step “1” on page 427 in “How the legacy PassTicket generator algorithm works” on page 427) left justified and padded to the right with hexadecimal '55's. For example, if the user ID is "TOM", PAD1 is 'E3D6D4555555' and PAD2 is '555555555555'. If the user ID is "IBMUSER", PAD1 is 'C9C2D4E4D2C5' and PAD2 is 'D95555555555'. PAD1 is used for time coder loop rounds 1, 3, and 5. PAD2 is used for time coder loop rounds 2, 4, and 6.

Step C

Result-B is encrypted using the RACF secured signon application key **2** as the encryption key to produce Result-C.

Step D

The left 2 bytes from the Result-C are isolated and the rest of the value is discarded, producing Result-D.

Step E

Result-D is XORed with L2B (from Result-A) to produce Result-E.

Step F

The values of L2B and R2B are redefined:

1. L2B is set equal to R2B.
2. R2B is set equal to Result-E.

Step G

R2B is permuted ³ using the permutation tables in [Figure 10](#) on page 429, where the table used reflects the number of the round. For example, for the first time through, R2B is permuted using table 1.

Step H

This step counts the number of time-coder rounds that have been completed. If the value is less than 6, the time-coder returns to Step b for another round. If 6 rounds have been completed, processing continues with the next step.

Step I

L2B (left 2 bytes) and R2B (right 2 bytes) are recombined into a 32-bit string. This completes the time-coder processing and produces Result-5. This result is passed back to the generator algorithm as input to Step “6” on page 427 for translation.

The legacy PassTicket permutation tables

A permutation table exists for each round of permutations that occurs during the legacy PassTicket time-coder process.

The six permutation tables work in the following manner:

- The upper of the two rows of numbers (O=>) represents the output positions, from left to right, of the 16 bits being permuted.
- The lower of the two rows (I=>) represents the input-bit position.

For example, using Permutation Table 1:

³ To permute is to transform or change the order of members of a group.

- Output-bit position 1 consists of the bit (on or off) in input-bit position 10.
- Output-bit position 2 consists of the bit in input-bit position 2.

Permutation Table 1																
O=>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I=>	10	2	12	4	14	6	16	8	9	1	11	3	13	5	15	7
Permutation Table 2																
O=>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I=>	1	10	3	12	13	16	7	15	9	2	11	4	5	14	8	6
Permutation Table 3																
O=>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I=>	3	10	1	12	13	16	9	15	7	2	14	4	5	11	8	6
Permutation Table 4																
O=>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I=>	10	4	12	2	14	8	16	6	9	1	13	3	11	5	15	7
Permutation Table 5																
O=>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I=>	4	10	12	1	8	16	14	5	9	2	13	3	11	7	15	6
Permutation Table 6																
O=>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I=>	1	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2

Figure 10. Permutation tables for RACF secured signon

The translation table

The translation table consists of 36 slots. The first 26 slots are occupied by the letters of the alphabet: A–Z. The last ten slots are occupied by the numerics: 0–9.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
26	27	28	29	30	31	32	33	34	35																
0	1	2	3	4	5	6	7	8	9																

Figure 11. Translation table for RACF secured signon

The legacy PassTicket translation process

The legacy PassTicket time-coder output produced by the process described in [Figure 7 on page 423](#) is translated into 8 alphanumeric characters in the following manner:

1. Bits 31, 32, 1, 2, 3, and 4 are translated to PassTicket character position 1, which is the leftmost position in the 8-byte alphanumeric PassTicket field.

To produce this character:

- The binary number, represented by the six bits, is divided by decimal 36.
- The remainder is used as an index into the translation table.

For example, a remainder of 0 translates to a PassTicket character of A and a remainder of 20 translates to a PassTicket character of U.

2. The process is repeated with the rest of the bit string:

- Bits 3 through 8 are translated to PassTicket character position 2.
- Bits 7 through 12 are translated to PassTicket character position 3.
- Bits 11 through 16 are translated to PassTicket character position 4.
- Bits 15 through 20 are translated to PassTicket character position 5.
- Bits 19 through 24 are translated to PassTicket character position 6.
- Bits 23 through 28 are translated to PassTicket character position 7.
- Bits 27 through 32 are translated to PassTicket character position 8.

Example

In this example, the result of the time-coder step is X'07247F79'. Using that value, a PassTicket is generated as follows:

Byte	1	2	3	4
Hexadecimal value	07	24	7F	79
Binary	00000111	00100100	01111111	01111001
Bit Position	00000000 12345678	01111111 90123456	11122222 78901234	22222333 56789012
PassTicket Character Position	Binary	Integer	Remainder	Translates to Character
1	010000	16	x 1/36 => 16	Q
2	000111	7	x 1/36 => 7	H
3	110010	50	x 1/36 => 14	O
4	100100	36	x 1/36 => 0	A
5	000111	7	x 1/36 => 7	H
6	111111	63	x 1/36 => 27	1
7	110111	55	x 1/36 => 19	T
8	111001	57	x 1/36 => 21	V

1. Bits 31, 32, 1, 2, 3, and 4 (6 bits total) are translated to produce the PassTicket character in position 1.
The six bits (binary '010000' or decimal 16) are divided by decimal 36.
2. The remainder (decimal 16) becomes the index into the translation table. The result is character 'Q'.
3. Repeat the process for the rest of the bits.
 - Bits 3 through 8 are translated to PassTicket character 'H'.
 - Bits 7 through 12 are translated to PassTicket character 'O'.
 - Bits 11 through 16 are translated to PassTicket character 'A'.
 - Bits 15 through 20 are translated to PassTicket character 'H'.
 - Bits 19 through 24 are translated to PassTicket character '1'.
 - Bits 23 through 28 are translated to PassTicket character 'T'.
 - Bits 27 through 32 are translated to PassTicket character 'V'.

The resulting PassTicket returned as output is QHOAH1TV.

How the enhanced PassTicket generator algorithm works

The RACF enhanced PassTicket generator algorithm uses the input information to create an enhanced PassTicket. By using cryptographic techniques, the algorithm ensures that each enhanced PassTicket is unpredictable. The enhanced PassTicket is an 8-character alphanumeric string which has a configurable character set. The PTKTDATA class profile can be configured to indicate the desired character set per application by using the TYPE keyword in the SSIGNON segment. The actual enhanced PassTicket depends on the input values.

The following steps describe this process:

1. The RACF user ID **1** and application name **3** are appended together to produce Result-1.
2. An HMAC with key **2** is performed on Result-1 to produce Result-2.

Note: All enhanced PassTicket cryptographic operations use HMAC with SHA-512, which produces 64 bytes of output.
3. The left 6 bytes from Result-2 are selected as input to the next step as Result-3. The rest are discarded.
4. Result-3 is XORed with the time and date information **4** to produce Result-4.
5. Result-4 is passed to the enhanced PassTicket time-coder routine to produce Result-5.

6. Result-5 from the time-coder routine is converted to an 8-character string called the enhanced PassTicket. Refer to [“How the enhanced PassTicket character conversion works”](#) on page 432.

How the enhanced PassTicket time-coder algorithm works

The RACF enhanced PassTicket time-coder algorithm uses the Result-4 from Step [“4”](#) on page 427 of the enhanced PassTicket generator algorithm. It creates the time-coder information Result-5 and passes it back to step [“6”](#) on page 427 of that algorithm.

The following steps, which make up Step [“5”](#) on page 427 of the enhanced PassTicket generator algorithm, describe this process:

Step A

Separate the 6-byte time-coder input (Result-4) into two portions, L3B (the left 3 bytes), and R3B (the right 3 bytes) to produce Result-A.

Step B

Concatenate R3B (the right 3 bytes from Result-A) with 17 bytes of padding bytes to form Result-B. In the resulting 20-byte string, the 3 bytes of R3B occupy the leftmost 3 byte positions.

The padding is a 17-byte string containing three separate fields:

1. The 1-byte round counter

The round counter starts with the value 1 and is incremented by 1 on each subsequent use.

2. The 8-byte user ID **1**

3. The 8-byte application name **3**

Step C

An HMAC with key **2** is calculated on Result-B to produce Result-C.

Step D

The left 3 bytes from the Result-C are isolated and the rest of the value is discarded, producing Result-D.

Step E

Result-D is XORed with L3B (from Result-A) to produce Result-E.

Step F

An enhanced PassTicket type MIXED is encoded as a 48-bit value and a type UPPER is encoded as a 41-bit value. This step sets the extraneous leftmost 7 bits of a type UPPER to binary zero.

When the enhanced PassTicket type is UPPER and the round counter is 1, 3 or 5, the following masking operation is performed:

- Perform bitwise AND on the leftmost 1 byte of Result-E with ‘01’x to set the leftmost 7 bits to zero to produce Result-F.

When the enhanced PassTicket type is MIXED or the type is UPPER and the round counter is 2, 4 or 6:

- Result-E is set as Result-F without any changes.

Step G

The values of L3B and R3B are redefined:

1. L3B is set equal to R3B.
2. R3B is set equal to Result-F.

Step H

This step counts the number of time-coder rounds that have been completed. If the value is less than 6, the time-coder returns to Step B for another round. If 6 rounds have been completed, processing continues with the next step.

Step I

L3B (left 3 bytes) and R3B (right 3 bytes) are recombined into a 48-bit string. This completes the time-coder processing and produces Result-5. This result is passed back to the generator algorithm as input to Step “6” on page 427 for translation.

The enhanced PassTicket translation table

The enhanced PassTicket translation table consists of 64 slots. The first ten slots are occupied by the numerics: 0–9. The next 26 slots are occupied by the uppercase letters of the alphabet: A–Z. The next 26 slots are occupied by the lowercase letters of the alphabet: a–z. The last two slots are occupied by the special characters: dash “-” and underscore “_”.

Note: An enhanced PassTicket with type UPPER will only use the first 36 slots (0-35) of this translation table.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
Q	R	S	T	U	V	W	X	Y	Z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
52	53	54	55	56	57	58	59	60	61	62	63														
q	r	s	t	u	v	w	x	y	z	-	_														

Figure 12. Translation table for RACF secured signon

How the enhanced PassTicket character conversion works

The RACF enhanced PassTicket time-coder output is converted to an EBCDIC string value using the following process:

Step A

Copy the 6-byte time-coder output value from Result-5 to the rightmost 6 bytes of a 64-bit binary value to produce Result-A. The leftmost 2 bytes of Result-A are set to binary zero.

Step B

For enhanced PassTicket type UPPER:

- Calculate modulo 36 of Result-A to produce Result-B.

For enhanced PassTicket type MIXED:

- Calculate modulo 64 of Result-A to produce Result-B.

Step C

Translate Result-B from a binary value to an EBCDIC value using the enhanced PassTicket type translation table to produce Result-C.

For example, the binary value 33 is translated to the EBCDIC value ‘X’.

Step D

Result-C is set as an individual character of the EBCDIC enhanced PassTicket value. The characters are concatenated together one at a time starting with the rightmost character and proceeding to the left on each round of conversion.

Step E

When less than 8 characters have been converted:

- For enhanced PassTicket type UPPER:
 - Divide Result-A by 36 to produce Result-E.
- For enhanced PassTicket type MIXED:
 - Divide Result-A by 64 to produce Result-E.

Step F

Replace Result-A with Result-F.

Step G

This step counts the number of characters that have been encoded. When there are less than 8 characters encoded the conversion process returns to **Step B** for another round.

Step H

The final enhanced PassTicket value has been assembled.

Generating a secured signon session key

Note:

1. IBM recommends that the secured signon session key *not* be used outside of a test environment. It is no longer considered secure. This section is left for reference only.
2. Enhanced PassTickets and enhanced PassTicket keys cannot be used to generate a secured signon session key.

An attempt to generate a secured signon session key with a specified enhanced PassTicket value may fail with Return Code 4 – “Incorrect PassTicket”

An attempt to generate a secured signon session key with a PTKTDATA class profile that contains only an enhanced PassTicket key may fail with Return Code 24 – “Error in the session key generator process”.

RACF can be invoked to generate a secured signon session key. A secured signon session key is a 64-bit value that can be used as a short-term session masking key for enhancing communication security between two network entities. The 64-bit value is a commercial data masking facility (CDMF) key which has an effective cryptographic strength of 40 bits. Because these keys are not highly secure, it is important that they be used only for communications of short duration.

Assume that a non-RACF z/OS application wants to communicate with a second party network entity. The application calls the secured signon session key generator service to create a secured signon session key based on a previously created PassTicket. The second party network entity uses the same algorithm and PassTicket to generate the same session key. The two network entities can now communicate securely without ever exchanging session keys.

Using the service to generate a secured signon session key

To allow RACF to create a secured signon session key, the non-RACF z/OS application calls the secured signon session key generator service.

The secured signon session key generator service:

- Is branch-entered by callers
- Is *not* supported in cross-memory mode
- Requires that the caller be in task mode, system key zero (0), and primary ASC mode

Before calling the secured signon session key generator service, the application must locate the address of the service. This address can be found in field RCVTSKGN in the RACF communications vector table, RCVT. The ICHPRCVT macro maps the RCVT and field CVTRAC points to it in the MVS communications vector table (CVT).

How the secured signon session key generator service works

The service:

- Uses standard linkage
- Uses the PassTicket as input for the algorithm
- Returns the session key in general purpose register 0 (4 bytes) and general purpose register 1 (4 bytes)
- Provides return codes

Notes:

1. The secured signon session key generator service uses either the current task level or address space level ACEE unless an ACEE address is passed on the input parameter list.

If an application is using a RACF PassTicket to authenticate users and wants to derive a session key for securing application-to-user communication, the application must establish a task level ACEE for its client or point to the client's ACEE. The following calls must be made **in this sequence**:

- a. A RACROUTE REQUEST=VERIFY,ENVIR=CREATE request to authenticate and create a task level ACEE for the application's client. (This request can be omitted if the client's ACEE was previously created by a RACROUTE REQUEST=VERIFY.)
 - b. Construct a secured signon session key generator parameter list and branch to the address pointed to by RCVTSKGN.
2. Register 13 points to a standard save area.
 3. No additional recovery processing is provided by the secured signon session key generator service beyond what is already in effect for the invoking program.

Invoking the secured signon session key generator service

Following is an example of a generalized programming technique you can use with assembler language to invoke this service. It is not intended to be syntactically correct.

```
LA 1,MY_APPL_PLIST
L 15,RCVTSKGN
CALL (15),(1)
```

Register 1 points to MY_APPL_PLIST which contains:

Displacement	Description
+0	A pointer to the RACF PassTicket used for user authentication
+4	A pointer to a one-byte length field followed by up to 8 characters which is the APPLID
+8	A pointer to the address of the user ID's ACEE that was created during PassTicket evaluation. If the address is zero, the task level ACEE (TCBSENV) is used if it exists. If not, the address space level ACEE (ASXBSENV) is used.

Return codes from the secured signon session key generator service

The secured signon session key generator service produces the following return codes in register 15:

Note: The values shown are in hexadecimal.

Return Code	Description
0	Successful completion. The resulting session key is contained in general purpose registers 0 and 1.
4	Incorrect PassTicket
8	No PTKTDATA profile found for the application
C	No task or address space ACEE found, and the ACEE pointer was not specified on the input parameter list.
10	Caller is not authorized
14	The RACF PTKTDATA class is not active
18	Error in the session key generator process

Incorporating the secured signon session key generator algorithm into your program

To generate a secured signon session key without using the secured signon session key generator service, you need to incorporate the secured signon session key generator algorithm into your program.

In order to ensure identical session key generation on both platforms, the following steps must be implemented by both the non-RACF application and the second party network entity.

The secured signon session key generator algorithm consists of two parts:

- Secured signon session key generation logic
- CDMF key-weakening logic

The flowcharts in Figure 13 on page 435 and Figure 14 on page 436 and the descriptions that follow show how to implement the secured signon session key generator algorithm.

Secured signon session key generation logic

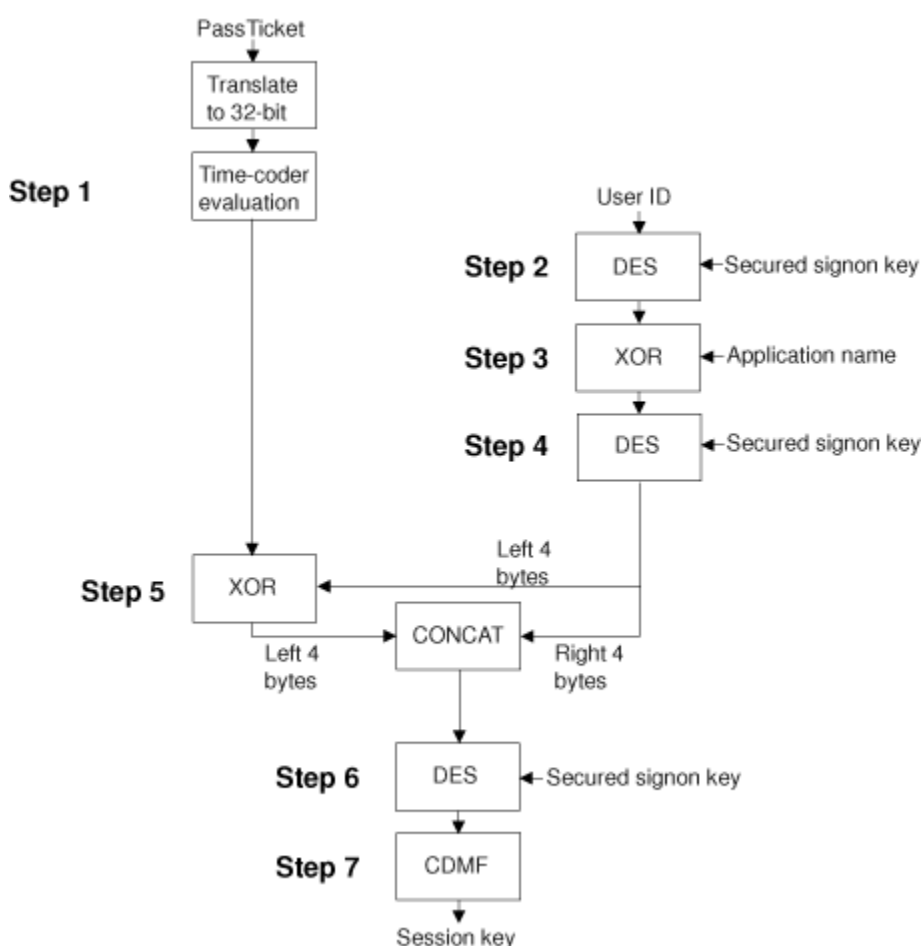


Figure 13. Secured signon session key generation logic

The secured signon session key generation logic is:

1. The PassTicket used to establish the session is time-coder evaluated to extract the time stamp.

This is the reverse process of steps 5 and 6 as described in [“Incorporating the PassTicket generator algorithm into your program” on page 421](#). If the time stamp used to generate the PassTicket is already known, this step can be skipped.

2. The input user ID is DES-encrypted with the secured signon key shared with the second party network entity.

Note: Steps 2 through 4 of this secured signon session key generation logic are the same as steps 1 and 2 of “Incorporating the PassTicket generator algorithm into your program” on page 421.

3. The result of step 2 is XORed with the non-RACF application name.
4. The result of step 3 is again DES-encrypted with the secured signon key.
5. The left 4 bytes of the result of step 4 are XORed with the left 4 bytes of the time stamp (result of step 1) and then concatenated with the right 4 bytes of the result of step 4.
6. The result of step 5 is DES-encrypted with the secured signon key to produce a strong session key.
7. The result of step 6 is weakened using CDMF to produce the final secured signon session key.

CDMF key-weakening logic

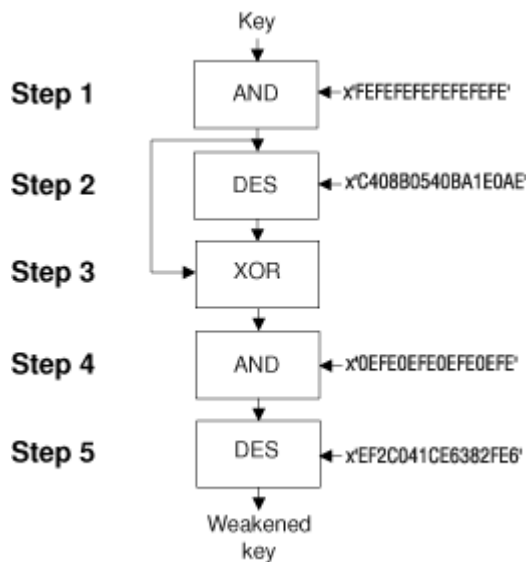


Figure 14. CDMF key-weakening logic

The CDMF key-weakening logic is:

1. The parity bits of the key are zeroed by ANDing it with the string X'FEFEFEFEFEFEFEFE'.
2. The result of step 1 is DES-encrypted with the key X'C408B0540BA1E0AE'.
3. The result of step 2 is XORed with the result of step 1.
4. The result of step 3 is ANDed with the string X'0EFE0EFE0EFE0EFE'.
5. The result of step 4 is DES-encrypted with the key X'EF2C041CE6382FE6' to produce the weakened key.

Chapter 11. The RACF environment service

RACF provides the environment service, IRRENS00, that is located through the RCVT. An RCVT flag, RCVTENVS, indicates that the service is present. When the flag is on, RCVTENVP contains the address of IRRENS00.

Note: This service is to be used only by z/OS UNIX System Services and is not intended for customer use. Information about this service is provided here to assist security vendors in understanding IRRENS00 operations.

Function

Keep-Controlled

Verifies that the environment is currently controlled, and if so sets flags indicating that it must remain controlled. It also saves a message supplied by the caller indicating the reason that the environment must remain controlled. It keeps separate flags for z/OS UNIX requests and RACF requests.

Mark-Uncontrolled

Determines whether the environment must remain controlled by inspecting the keep-controlled indicators and (if necessary) the PADS data set list from the ACEE and the CDEs for modules in the address space, and if not, marks it uncontrolled and also marks any existing TCBs and CDEs as uncontrolled. However, if a keep-controlled request is outstanding, mark-uncontrolled returns an error code and either issues (WTO) any saved messages from a previous keep-controlled request or generates messages to indicate why RACF needed the environment kept controlled. For a successful request, mark-uncontrolled also saves a caller-provided message indicating why the environment became uncontrolled, which it issues on subsequent keep-controlled requests if necessary.

Reset Keep-Controlled

Resets the keep-controlled indicators and removes any saved messages relating to previous keep-controlled requests. It is intended only for use by z/OS UNIX when z/OS UNIX determines that the environment no longer needs to be kept controlled.

Query

Returns a reason code indicating the state of the environment.

Requirements

1. Authorization: Both supervisor state and key 0.
2. Dispatchable Unit Mode: Task.
3. Cross-Memory Mode: PASN=HASN=SASN.
4. AMODE: 31.
5. RMODE: Any.
6. ASC Mode: Primary.
7. Recovery Mode: Estae (caller cannot have an FRR).
8. Serialization: This service obtains Local lock and uses compare and swap for serialization.
9. Control Parameters: The parameter list and all parameters must be in the primary address space.

RACF authorization

None

Register usage

Registers on input:

- 1** Parameter list address
- 13** Save area address
- 14** Return address
- 15** Address of IRRENS00 from RCVTENVP

Registers on return:

- 0** Reason code
- 1–12** Unknown
- 13** Save area address
- 14** Return address
- 15** Return code

Format

```

*      .
*      .
*      .
      L    Rx,CVTPTR
      USING CVT,Rx
      L    Rx,CVTRAC
      USING RCVT,Rx
      TM    RCVTMFL1,RCVTENVS    Test for service availability
      BZ    not_available        bypass call if not available
      L    R15,RCVTENVP          Get service address
      LA    R1,parms             Get parameter list area
      L    R13,save_area         Must provide a save area
      BALR  R14,R15              Call the service
*      .or CALL (15),(work_area,function_code,function_flags,return_code,
*      .                      reason_code,message_block)
      DROP  Rx
      ...
parms DS    0F
      DC    A(work_area)
      DC    A(function_code)
      DC    A(function_flag)
      DC    A(return_code)
      DC    A(reason_code)
      DC    A(message_block)
      ...
work_area DS    0D,XL2048
function_code DS    F
function_flags DS    F
return_code DS    F
reason_code DS    F
message_block DS    0F
message_text DS    F'length',CL(length)'text'
*      .

```

Parameters

work_area

2048 bytes, doubleword aligned. Used as internal storage by IRRENS00.

function_code

Fullword; input:

X'00000000'

Keep-Controlled

X'00000001'

Mark-Uncontrolled

X'00000002'

Reset Keep-Controlled

X'00000003'

Mark file system

X'00000004'

Query

function_flag

Fullword; input:

For Mark-Uncontrolled:

X'00000000'

IRRENS00 should issue WTO on failure.

X'00000001'

IRRENS00 should not issue WTO.

X'80000000'

Request by z/OS UNIX for file system control.

For Keep-Controlled:

X'80000000'

Request by z/OS UNIX.

X'40000000'

Request by RACF.

Note:

One, and only one, of the above flags must be on.

X'00000000'

IRRENS00 should issue WTO on failure.

X'00000001'

IRRENS00 should not issue WTO.

X'00000002'

Only z/OS UNIX mark-uncontrolled should be checked before marking keep-controlled.

X'00000004'

Indicates that RACF should determine whether ENHANCED program security is in effect for this job, and if so RACF should determine whether the execution environment was established by a MAIN program or not. If not, RACF either fails the request or sets a warning with return and reason codes. Additionally, RACF should issue appropriate messages, including those saved from a prior call to IRRENS00.

For Reset Keep-Controlled:

X'80000000'

Reset the z/OS UNIX keep-controlled flag and clear the related saved message.

X'40000000'

Reset the RACF keep-controlled flag and clear the related saved message.

X'20000000'

Clear any message saved for a previous mark-uncontrolled function, and reset the z/OS UNIX mark-uncontrolled flag.

Note: At least one of the above flags must be on.

For Mark file system:

X'80000000'

Indicates a call from z/OS UNIX System Services.

Note: The above flag must be on.

For Query:

X'00000000'

Both RACF state and z/OS UNIX state are checked.

X'00000002'

Only z/OS UNIX state is checked; RACF state is not checked.

return_code

Fullword; output. See **Note** below.

reason_code

Fullword; output. See **Note** below.

message_block

Fullword length of text, followed by the text input, or zero for reset keep-controlled and query.

Note: The fullword must be nonzero and message text must be provided for the keep-controlled and mark-uncontrolled functions. The maximum message text length allowed is 250. The fullword must be 0 for the reset keep-controlled and query functions.

For keep-controlled and mark-uncontrolled, the caller uses this parameter to provide a message indicating the reason for the request. The first character of the message must not be a blank. The message must begin with a message ID (such as BPXnnnI or ICHnnnI), followed by a blank. When the message is displayed, a maximum of 71 characters appear on each line of the display. If a blank does not appear in column 71 or column 72, the message text is split to a new line at the preceding blank. The caller providing the message must document it. Message text must be uppercase, must contain only alphanumeric and national characters, and should be NLS-compliant.

A copy of the message provided by the caller is saved and IRRENS00 issues this message, among others, during failing keep-controlled requests or failing mark-uncontrolled requests. The messages are issued with descriptor code 6 and routing codes 9 and 11, directing them to the security console and the programmer.

The caller is not responsible for deleting the saved copy of the messages. Saved messages are cleared and their storage released by the Reset Keep-Controlled function when the corresponding keep-controlled indicator is reset. IRRENS00 is responsible for ensuring that the storage used to contain the saved messages is released appropriately when no longer needed.

Return and reason codes

IRRENS00 returns the following return and reason codes:

Return code**Meaning**

0

Function successful

Reason code

0

4

If Query, the environment is dirty.

8

If Query, the environment is clean but can be made dirty without error.

C

If Query, the environment must stay clean.

20

Indicates that a caller requested a MAIN check through function flag X'00000004', that enhanced program security is in effect in warning mode for the current job, and the current execution environment was not established by a MAIN program.

4

Function not processed; parameter error

Reason code**Meaning****4**

Incorrect function code

8

Incorrect function flags for specified function code

C

Incorrect message block for specified function code

8

Function failed

Reason code**Meaning****4**

Cannot mark keep-controlled for z/OS UNIX; already uncontrolled

8

Cannot mark keep-controlled for RACF; already uncontrolled

C

Cannot mark uncontrolled; marked keep-controlled for z/OS UNIX

10

Cannot mark uncontrolled; marked keep-controlled for RACF

20

Indicates that a caller requested a MAIN check through function flag X'00000004', that enhanced program security is in effect in failure mode for the current job, and the current execution environment was not established by a MAIN program.

Note: Keep-controlled can only fail due to the environment already being marked uncontrolled. Likewise, mark-uncontrolled can only fail because of a previous keep-controlled request. For a mark-uncontrolled request, the z/OS UNIX keep-controlled request is checked first, and if on, return code 8 and reason code C are returned, without additional checking for RACF keep-controlled.

C

Internal error

Reason code**0**

Usage notes

1. Callers (including z/OS UNIX) can use the IRRENS00 service after checking that RCVTENVS='1'B. The address for IRRENS00 is obtained from RCVTENVP.
2. To ensure that a message is saved, the mark-uncontrolled function of IRRENS00 should be used rather than setting TCBNCTL directly.
3. Only IRRENS00 sets, checks, or resets the keep-controlled indicators. Resetting can occur when the reset function is requested, or at other times if IRRENS00 determines that the environment no longer needs to be kept controlled. For example, during a mark-uncontrolled request IRRENS00 might find RACF's keep-controlled indicator set, but find no open program-accessed data sets and no execute-

controlled modules present. In this case, it turns off RACF's keep-controlled indicator, and if the z/OS UNIX indicator is off, it honors the request.

4. The z/OS UNIX mark-uncontrolled indicator is kept internally by the security product, and is set in addition to TCBNCTL. Defining BPX.DAEMON.HFCTL in the FACILITY class requests that z/OS UNIX enforce file system control only. This option is appropriate when the loading of uncontrolled files must be restricted to protect against changes made by superusers, but the loading of uncontrolled programs from MVS libraries does not introduce any security concerns.

With file system control in effect, z/OS UNIX passes the z/OS UNIX mark-uncontrolled indicator to IRRENS00 when an uncontrolled file is loaded from the file system. The message passed with the first z/OS UNIX mark-uncontrolled request is always saved.

With file system control in effect, z/OS UNIX passes the check z/OS UNIX mark-uncontrolled indicator to IRRENS00 on a keep-controlled request indicating that the request to keep-controlled should only fail if z/OS UNIX marked the environment uncontrolled.

The z/OS UNIX mark-uncontrolled indicator is reset when the messages saved for previous mark-uncontrolled requests are cleared by the reset function.

Related services

None

Chapter 12. SAF user mapping plug-in interface

SAF defines a service that z/OS applications can use to retrieve the SAF user ID corresponding to a specified user credential from another registry. A default implementation or installation-provided "plug-in" routines to this service provide the credential mapping logic, which can be unique to specific applications or security products.

The SAF user mapping plug-in interface is a C or C++ programming interface that consists of function prototypes and a default implementation that can be replaced by customers or vendors. This plug-in interface enables the user ID mapping capabilities by providing the following functions:

- safMappingInit()
- safMappingLookup()
- safMappingTerm()

Installation considerations for the SAF user mapping service

The SAF user mapping plug-in interface is used by C and C++ applications. Developers of applications that use or implement the plug-in interface should use the SIEAHADR(IRRSPIM) or /usr/include/irrspim.h header file to define the plug-in interface. The following table lists the shipped parts of the SAF user mapping plug-in implementation.

Table 293. SAF user mapping plug-in dlls, header files, and side deck		
Data set member	HFS (hierarchical file system)	Description
SIEALNKE(IRRSPIM)		SAF user mapping interface module
SIEALNKE(IRRSPIME)		Default plug-in module
SIEAHDR(IRRSPIM)	/usr/include/irrspim.h	Header file that defines the SAF user mapping interface
SIEASID(IRRSPIM)	/usr/lib/irrspim.x	Definition side file for the SAF user mapping interface

The DLLs, IRRSPIM and IRRSPIME in SIEALNKE need to be program-controlled. See *z/OS Security Server RACF Security Administrator's Guide* for details on how to set up program control.

The default implementation of the plug-in interface uses the mappings that are stored in an EIM (Enterprise Identity Mapping) domain. The implementation requires access to eim.dll and ldap.dll, which are linked to /usr/lib. See *z/OS Integrated Security Services EIM Guide and Reference* for more information.

Writing an application that uses the SAF user mapping plug-in interface

A C/C++ application calls the SAF user mapping plug-in interface to retrieve the local z/OS identity for a user who was authenticated on a different platform. The interface consists of three services that perform the following functions:

- Initialize the connection with the plug-in implementation. This function is invoked once.
- Search in a repository of user ID mappings for a mapping between the source user credential and the local SAF user credentials. This function is invoked as many times as needed.
- Close the connection with the plug-in implementation. This function is invoked once.

The following diagram illustrates the interactions between a C/C++ application, the DLL (dynamic link library) of the SAF user mapping interface, and the default plug-in implementation.

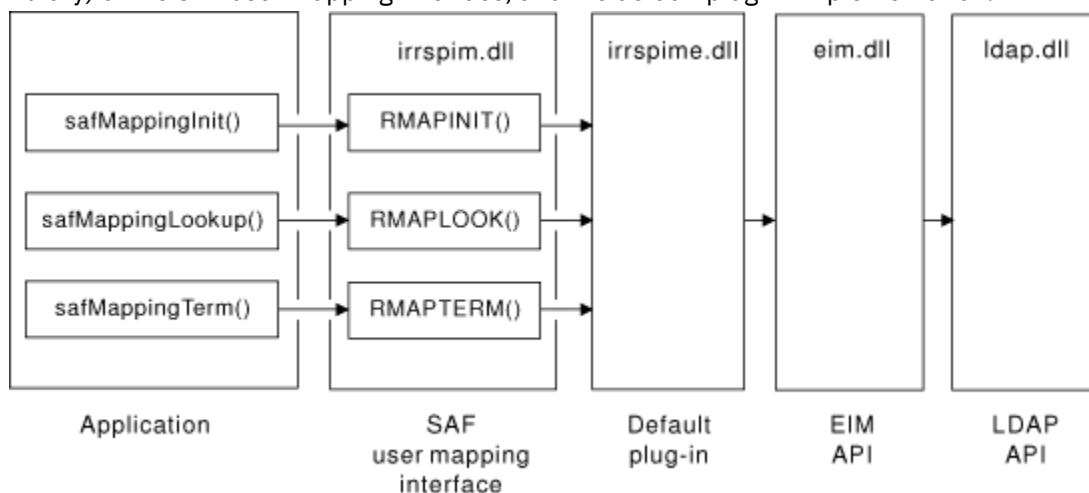


Figure 15. SAF user mapping using EIM

The following example is a fragment of a C application using the plug-in implementation.

```
#pragma runopts(POSIX(ON))
#include <stdio.h>
#include <irrspim.h>

void printmErr(SafmapErr *);

int main()
{
    int mRc = 0;

    char      * aData   = NULL;
    char      * dllName = NULL;
    int        i         = 0;
    SafmapHandle * mh    = NULL;
    SafmapErr  * mErr    = NULL;
    SafmapErr  mErrData;
    SafmapHandle mHandle;
    SafmapResult * mResult = NULL;
    char      mResultData[sizeof(SafmapResult) + 8 + 1];
    SafmapCreds * sUser   = NULL;
    SafmapCreds  sUserData;

    /*-- Initialize the plug-in -----*/
    bzero((char *) &mHandle, sizeof(SafmapHandle));
    mh = &mHandle;
    dllName = SAFMAP_DEFAULT_PLUGIN;
    bzero((char *) &mErrData, sizeof(mErrData));
    mErr = &mErrData;

    mRc = safMappingInit (mh, dllName, mErr);

    if (mRc > 0) {
        fprintf(stderr, "safMappingInit() failed. rc =%i\n", mRc);
        printmErr(mErr);
        return (-1);
    }

    /*-- Lookup a user mapping -----*/
    bzero((char *) &sUserData, sizeof(SafmapCreds));
    sUser = &sUserData;
    sUser->credsType = SAFMAP_REGISTRY_USER;
    sUser->credsCCSID = SAFMAP_DEFAULT_CCSID;
    sUser->credsData.RegistryUser.registry = "My Kerberos Realm";
    sUser->credsData.RegistryUser.user = "kuser";

    aData = NULL;

    bzero((char *) &mResultData, sizeof(mResultData));
    mResult = (SafmapResult *) &mResultData;
    mResult->bytesAvailable =
        sizeof(mResultData) - sizeof(SafmapResult) + 1;
}
```

```

mResult->resultCreds.credsType = SAFMAP_USER_ONLY;

bzero((char *) &mErrData, sizeof(mErrData));

mRc = safMappingLookup(mh, sUser, aData, mResult, mErr);

switch (mRc) {
case 0:
    fprintf(stderr, "safMappingLookup() returned userid %s.\n",
        mResult->resultCreds.credsData.UserOnly.user);
    break;
case SAFMAP_WARNING:
    fprintf(stderr, "safMappingLookup() returned a warning.\n");
    break;
case SAFMAP_ERROR:
    fprintf(stderr, "safMappingLookup() returned an error.\n");
    break;
case SAFMAP_SEVERE:
    fprintf(stderr, "safMappingLookup() returned a severe error.\n");
    break;
}
printmErr(mErr);
if (mRc > 0)
    return (-1);

/*-- Close the connection with the plug-in -----*/
bzero((char *) &mErrData, sizeof(mErrData));

mRc = safMappingTerm(mh, mErr);

if (mRc > 0) {
    fprintf(stderr, "safMappingTerm() failed. rc = %i\n", mRc);
    printmErr(mErr);
    return (-1);
}
}

void printmErr( SafmapErr * mErr)
{
    fprintf(stderr, "mErr->mpiReturnCode %i\n", mErr->mpiReturnCode);
    fprintf(stderr, "mErr->mpiReasonCode %i\n", mErr->mpiReasonCode);
    fprintf(stderr, "mErr->mpiVersion %s\n", mErr->mpiVersion);
    fprintf(stderr, "mErr->mpiInfo %s\n", mErr->mpiInfo);
    fprintf(stderr, "mErr->message %s\n", mErr->message);
    return;
}

```

The application can be compiled from JCL or the z/OS UNIX System Services. The default plug-in implementation requires the execution environment to be running with POSIX(ON).

If the application that is calling the plug-in interface is in the z/OS UNIX shell, and POSIX is already set to ON, nothing needs to be done.

If the application is running from a batch job or TSO, POSIX must be explicitly turned on. There are a number of methods available. The sample code shows the use of the `#pragma runopts`. Another option is to specify the runtime option as an LE (Language Environment®) parameter on the `exec` statement. For detailed information about how to use the LE parameters, see *z/OS Language Environment Programming Guide*.

Preparing to run an application with the SAF user mapping plug-in implementation

Before the application can run using the default SAF user mapping plug-in implementation, EIM and RACF must be set up with the information used by that plug-in.

Note: Other SAF user mapping plug-in implementations might require a different setup. See [“Writing your own SAF user mapping plug-in implementation” on page 446](#) for the details.

To set up EIM, perform the following steps.

1. Define and initialize the EIM domain with the user ID mappings or policies.

2. Define the RACF profiles containing the name of the LDAP server hosting the EIM domain, the name of the EIM domain, and the LDAP BINDDN and BINDPW that the plug-in implementation uses.
3. Define the IRR.PROXY.DEFAULTS profile in the FACILITY class, activate the class if needed and RACLIST the FACILITY class profile. This profile should contain the name given to the local z/OS registry in the EIM domain.

For details on how to perform these steps, see *z/OS Integrated Security Services EIM Guide and Reference* for information about the software requirements and setup instructions.

If the C/C++ application is unauthorized (that is, problem program state and problem program key), the caller of the default SAF user mapping plug-in implementation needs to have access to either of the following RACF resources in the FACILITY class.

- The user ID associated with the address space must have READ access to the BPX.SERVER resource in the FACILITY class.
- The current user ID must have READ access to the IRR.RDCEKEY and IRR.RGETINFO.EIM resources in the FACILITY class.

In addition, all unauthorized applications that use the default SAF user mapping plug-in implementation must run in a clean address space. Minimally, the unauthorized application must be program-controlled. See *z/OS Security Server RACF Security Administrator's Guide* for details on how to set up program control for the application.

Logging of the EIM lookup operations can also be performed by defining RACF profiles. See *z/OS Integrated Security Services EIM Guide and Reference* for the setup instructions.

Writing your own SAF user mapping plug-in implementation

The SAF user mapping interface accepts plug-in implementations other than the default one provided with z/OS.

The following code can be used as the starting point for an implementation.

```
/*--- Define the DLL's interface ----- */
#pragma export(RMAPINIT)
#pragma export(RMAPLOOK)
#pragma export(RMAPTERM)

/*--- Includes ----- */
#include <irrspim.h>

/*--- Constants ----- */
#define RMAPINIT_MSG "rmapinit called."
#define RMAPLOOK_MSG "rmaplook called."
#define RMAPTERM_MSG "rmapterm called."

/*--- RMAPINIT function ----- */
int RMAPINIT(
    SafmapHandle * mh,
    char * dllName, /* Ignored */
    SafmapErr * mErr)
{
    /* - Validate parameters. */
    /* o There will always be SafmapHandle and SafmapErr parameters. */
    /* o The dllName is provided for information purposes only. It may be NULL. */
    /* - Update the mapping handle with plug-in specific data if needed. The data can be anchored off the mapping handle. mpiData field in the SafmapHandle is provided for that purpose. */
    /* - Perform any initialization required by the plug-in implementation. */
    /* - Update the mapping error structure with the following: */
    /* o mErr->mpiReasonCode: A value with meaning specific to the plug-in implementation. RECOMMENDED. */
    /* o mErr->mpiVersion : A string that indicates the release of
```

```

/*                                     of the plug-in implementation.          */
/*                                     STRONGLY RECOMMENDED.                    */
/* o mErr->mpiInfo      : A string that contains any other                     */
/*                                     information that might assist             */
/*                                     callers in identifying configuration     */
/*                                     problems.                                */
/*                                     OPTIONAL.                                */
/* o mErr->message      : A string that describes the results                  */
/*                                     returned or error encountered.           */
/*                                     STRONGLY RECOMMENDED.                    */
/* - Return one of the SAF user mapping plug-in reason codes                  */
/* as the function return value.                                              */
/* Note: Be sure to release any resources when the return value              */
/* is SAFMAP_ERROR or SAFMAP_SEVERE. SAFMAP_ERROR should be                  */
/* used when re-initializing the connection will mostly likely              */
/* correct the problem. SAFMAP_SEVERE should be used when                   */
/* additional work is required by administrators or system                   */
/* programmers.                                                              */
return (SAF-user-mapping-plug-in-reason-code);
}

/*--- RMAPLOOK function ----- */
int RMAPLOOK(
    SafmapHandle * mh,
    SafmapCreds * sUser, /* Source user */
    char * aInfo, /* Application info */
    SafmapResult * mResult, /* Mapped result */
    SafmapErr * mErr)
{
    /* - Validate parameters. */
    /* o There will always be SafmapHandle, SafmapCreds, */
    /*   SafmapResult, and SafmapErr parameters. */
    /* o The application info parameter can be used to resolve */
    /*   any lookups that would otherwise find multiple z/OS user ids. */
    /*   This parameter may be NULL. */
    /* o Make sure the credential types are supported. */
    /* o Make sure the sUser->credentialCCSID is supported. */
    /* - Update the mapping handle with plug-in specific data if */
    /*   needed. */
    /* - Perform the lookup. */
    /* - Update the mapping error structure with the following: */
    /* o mErr->mpiReasonCode: A value with meaning specific to the */
    /*   plug-in implementation. */
    /*   RECOMMENDED. */
    /* o mErr->mpiVersion : A string that indicates the release of */
    /*   of the plug-in implementation. */
    /*   STRONGLY RECOMMENDED. */
    /* o mErr->mpiInfo : A string that contains any other */
    /*   information that might assist */
    /*   callers in identifying configuration */
    /*   problems. */
    /*   OPTIONAL. */
    /* o mErr->message : A string that describes the results */
    /*   returned or error encountered. */
    /*   STRONGLY RECOMMENDED. */
    /* - Return one of the SAF user mapping plug-in reason codes */
    /* as the function return value. */
    /* Note: Be sure to release any resources when the return value */
    /* is SAFMAP_ERROR or SAFMAP_SEVERE. SAFMAP_ERROR should be */
    /* used when re-initializing the connection will mostly likely */
    /* correct the problem. SAFMAP_SEVERE should be used when */
    /* additional work is required by administrators or system */
    /* programmers. */
    return (SAF-user-mapping-plug-in-reason-code);
}

/*--- RMAPTERM function ----- */
int RMAPTERM(
    SafmapHandle * mh,
    SafmapErr * mErr)
{
    /* - Validate parameters. */
    /* o There will always be SafmapHandle and SafmapErr parameters. */
    /* - Release storage and connections as needed. */
    /* - Update the mapping error structure with the following: */
    /* o mErr->mpiReasonCode: A value with meaning specific to the */
    /*   plug-in implementation. */
    /*   RECOMMENDED. */

```

```

/* o mErr->mpiVersion   : A string that indicates the release of   */
/*                       of the plug-in implementation.           */
/*                       STRONGLY RECOMMENDED.                   */
/* o mErr->mpiInfo       : A string that contains any other        */
/*                       information that might assist            */
/*                       callers in identifying configuration     */
/*                       problems.                                */
/*                       OPTIONAL.                                */
/* o mErr->message        : A string that describes the results    */
/*                       returned or error encountered.          */
/*                       STRONGLY RECOMMENDED.                   */
/* - Return one of the SAF user mapping plug-in reason codes     */
/*   as the function return value.                                */
/* Note: Be sure to release any resources when the return value  */
/*       is SAFMAP_ERROR or SAFMAP_SEVERE.                        */
/*
return (SAF-user-mapping-plugin-reason-code);
}

```

The functions RMAPINIT, RMAPLOOK, and RMAPTERM in the plug-in implementation correspond to the safMappingInit(), safMappingLook(), and safMappingTerm() functions. The parameter lists are the same and contain the same values.

SAF user mapping plug-in initialization function – safMappingInit()

Function

This function is used to initialize the SAF user mapping plug-in implementation.

Format

```

#include <irrspim.h>

int safMappingInit(
    SafmapHandle * mh,
    char          * dllName,
    SafmapErr     * mErr);

```

Requirements

1. Language: C or C++.
2. Authorization: Problem program or supervisor state, any key.
3. Dispatchable unit mode: Task.
4. Cross-memory mode: PASN=HASN.
5. AMODE: 31 bit.
6. RMODE: Any.
7. ASC Mode: Primary.
8. Serialization: Enabled for interrupts.
9. Locks: No locks held.
10. Control parameters: All storage must be in the primary address space.

RACF authorization

The safMappingInit() function does not require any special authorization. However, the default EIM plug-in implementation, which can be invoked by this function, does require the following authorization for the calling application:

- The calling application is running in system key or supervisor state, or
- The following conditions are met.
 - The calling application is program-controlled and the address space is clean.

- The FACILITY class is active and RACLISTed.
- The RACF user ID of the caller's address space has READ authority to the BPX.SERVER profile in the FACILITY class, or the current RACF user ID has READ authority to the IRR.RDCEKEY and IRR.RGETINFO.EIM profiles in the FACILITY class.

Note: Other SAF user mapping plug-in implementations might have different authorization requirements.

Usage notes

The safMappingInit() parameters behave as follows:

Table 294. The parameters of the safMappingInit() function		
Parameter	Input/output	Description
mh	Input/output	(Required) A pointer to a SafmapHandle structure. The mh handle is used by the plug-in interface and plug-in implementations to anchor data that must persist across the calls to the plug-in interface.
dllName	Input	(Optional) A pointer to a character string. The dllName parameter identifies the dynamic link library that implements the plug-in interface. The dllName parameter can be NULL or a character string. When it is NULL, the system plug-in, irrspime, is used. The dll name can be any name acceptable to the dlopen() service. See <i>z/OS C/C++ Runtime Library Reference</i> for details about the dlopen() service. The character string is encoded in the IBM-1047 CCSID (coded character set identifier).
mErr	Input/output	(Required) A pointer to a SafmapErr structure. The mErr structure contains more detailed information about the success or failure of the request to the plug-in interface. The mErr structure contains a return code, a reason code provided by the plug-in implementation, and an error string that gives more information about the results of the request. The maximum length of the error string is 256 bytes. The error string is NULL terminated. The mErr structure might also contain information about the version of the plug-in implementation and the plug-in implementation specific data such as the configuration settings. Any strings in the mErr structure are encoded in IBM-1047 CCSID.

The default plug-in implementation of the SAF user mapping interface uses the Enterprise Identity Mapping (EIM) services. EIM obtains its configuration information from RACF profiles. The name of the LDAP host name, the bind distinguished name (binddn), bind password (bindpw), and the EIM domain name can be stored in the following locations.

- An LDAPBIND class profile associated with the caller's user profile
- The IRR.EIM.DEFAULTS profile in the LDAPBIND class
- The IRR.PROXY.DEFAULTS profile in the FACILITY class

The name given to the local SAF registry in EIM is stored in the IRR.PROXY.DEFAULTS profile. See *z/OS Integrated Security Services EIM Guide and Reference* for guidance on how to set up these profiles.

Function return values

The `safMappingInit()` function returns a number of pieces of information to help you with problem determination. The SAF return value, the plug-in return code (`mErr->mpiReturnCode`), the plug-in reason code (`mErr->mpiReasonCode`), and an error string (`mErr->message`) are returned by the plug-in interface or the plug-in implementation. In addition, the version of the plug-in interface (`mErr->mpiVersion`) and any plug-in implementation specific data (`mErr->mpiInfo`) can be found in the `SafmapErr` structure.

Special processing occurs for parameter errors (`mErr->mpiReturnCode == SAFMAP_ERROR_PARMERR`) and unsupported credential types (`mErr->mpiReturnCode == SAFMAP_ERROR_NOTSUP`). The plug-in reason code (`mErr->mpiReasonCode`) contains a number identifying which parameter in the parameter list is in error.

The cleanup of the connection with the plug-in implementation occurs when the SAF return value is `SAFMAP_WARNING`, `SAFMAP_ERROR` or `SAFMAP_SEVERE`.

The SAF return values and the plug-in return codes (`mErr->mpiReturnCode`) are listed in the following table. These SAF return values and plug-in return codes are standard across the plug-in implementations. The plug-in reason code (`mErr->mpiReasonCode`) is unique to the plug-in implementation, except as previously noted.

Table 295. The SAF return values and the plug-in reason codes for the <code>safMappingInit()</code> function		
SAF return value	Plug-in return code (<code>mErr->mpiReturnCode</code>)	Explanation
0	0	Success
SAFMAP_WARNING (4)	SAFMAP_ERROR_NOTSUP (24)	One of the parameters contains a value that is not supported. Check <code>mErr->mpiReasonCode</code> to identify the parameter that contains the unsupported value.
SAFMAP_SEVERE (12)	SAFMAP_ERROR_INTERFACE(16)	The plug-in interface detected a problem internal to the <code>irrspim</code> dll.
SAFMAP_SEVERE (12)	SAFMAP_ERROR_PARMERR (28)	The parameter list for the plug-in contains an error. Check <code>mErr->mpiReasonCode</code> to identify the parameter that is in error.
SAFMAP_SEVERE (12)	SAFMAP_ERROR_PLUGIN (32)	An error internal to the plug-in implementation occurred.
SAFMAP_SEVERE (12)	SAFMAP_ERROR_SETUP (36)	The plug-in detected a problem in how the plug-in implementation is configured.

SAF user mapping plug-in lookup function – `safMappingLookup()`

Function

This function returns SAF user credentials when it can find a mapping from the source user credentials to a SAF user. The source user information and the returned user information can be a user ID, a user ID and password, or some other forms of user credentials. The plug-in implementation might be able to process all types of credentials or just a subset. The default plug-in implementation accepts a source user ID and returns a SAF user ID.

The first call to `safMappingLookup()` must be preceded by a call to `safMappingInit()`.

Format

```
#include <irrspim.h>

int safMappingLookup(
    SafmapHandle * mh,
```



```

SafmapCreds * sUser,
char * aData,
SafmapResult * mResult,
SafmapErr * mErr
);

```

Requirements

1. Language: C or C++.
2. Authorization: Problem program or supervisor state, any key.
3. Dispatchable unit mode: Task.
4. Cross-memory mode: PASN=HASN.
5. AMODE: 31 bit.
6. RMODE: Any.
7. ASC Mode: Primary.
8. Serialization: Enabled for interrupts.
9. Locks: No locks held.
10. Control parameters: All storage must be in the primary address space.

RACF authorization

The calling application does not require special authorization before calling `safMappingLookup()` when you use the default SAF user mapping plug-in implementation. However, the calling application requires authorization to create a mapping handle. See [“SAF user mapping plug-in initialization function – `safMappingInit\(\)`” on page 448](#) for more information.

Note: Other SAF user mapping plug-in implementations might have different authorization requirements.

Usage notes

The `safMappingLookup()` parameters behave as follows:

Table 296. The parameters of the <code>safMappingLookup()</code> function		
Parameter	Input/output	Description
<i>mh</i>	Input/output	(Required) A pointer to a <code>SafmapHandle</code> structure. The <i>mh</i> handle is used by the plug-in interface and implementations to anchor data that must persist across the calls to the plug-in interface.

Table 296. The parameters of the <i>safMappingLookup()</i> function (continued)		
Parameter	Input/output	Description
<i>sUser</i>	Input	<p>(Required) A pointer to a SafmapCreds structure.</p> <p><i>sUser</i> contains the credentials for the known or source user. The credentials are assumed to be NULL terminated.</p> <p>A number of credential types can be used with the plug-in, however, not all plug-in implementations support all possible types. Documentation for the plug-in implementations describes the types that are recognized by the implementation.</p> <p>The default plug-in implementation recognizes credentials that are made up of an EIM registry name and a user name (credential type SAFMAP_REGISTRY_USER).</p> <p>The default plug-in implementation accepts character strings encoded in the IBM-1047 CCSID.</p> <p>Other plug-in implementations might support additional CCSIDs.</p>
<i>aData</i>	Input	<p>(Optional) A NULL terminated character string.</p> <p>The <i>aData</i> parameter contains additional information that a plug-in implementation can use to refine the selection of the SAF user credentials. This information is useful when there are multiple mappings from one set of source user credentials to several SAF user credentials. This pointer can be NULL. There are no default values.</p> <p>The default SAF user mapping plug-in implementation can use this value. However, plug-in implementations are not required to support it. See <i>z/OS Integrated Security Services EIM Guide and Reference</i> for information about how this string can be used with the <i>eimGetTargetFromSource()</i> API.</p> <p>The default plug-in implementation accepts character strings encoded in the IBM-1047 CCSID.</p> <p>Other plug-in implementations might support additional CCSIDs.</p>

Table 296. The parameters of the `safMappingLookup()` function (continued)

Parameter	Input/output	Description
<i>mResult</i>	Input/output	<p>(Required) A pointer to a <code>SafmapResult</code> structure that is immediately followed by storage containing the returned credentials.</p> <p>The <code>mResult</code> structure contains the SAF user credentials. A number of credential types can be returned; however, not all plug-in implementations support all possible types. Documentation for the plug-in implementation describes the types that are recognized by the implementation.</p> <p>The default SAF user mapping plug-in implementation returns a SAF user ID (<code>SAFMAP_USER_ONLY</code>).</p> <p>The <code>bytesAvailable</code> field contains the length of the storage following the <code>SafmapResult</code> structure. When the credential type is <code>SAFMAP_USER_ONLY</code>, the recommended value is 9, that is, eight bytes for the maximum length of the SAF user ID and one byte for the NULL terminator.</p> <p>The <code>credentialLen</code> field is set to the length of the credential found during the mapping lookup. It includes the NULL terminators in the length.</p> <p>A credential is returned when it meets the criteria set by the plug-in implementation.</p> <p>For the default plug-in implementation, a credential is returned when it is no longer than eight characters. The pointers in the credential are set to NULL when a credential is not returned.</p> <p>The default plug-in implementation only returns credentials encoded in the IBM-1047 CCSID.</p> <p>Other plug-in implementations might support additional CCSIDs.</p>
<i>mErr</i>	Input/output	<p>(Required) A pointer to a <code>SafmapErr</code> structure.</p> <p>The <code>mErr</code> structure contains more detailed information about the success or failure of the request to the plug-in interface.</p> <p>The <code>mErr</code> structure contains a return code, a reason code provided by the plug-in implementation, and an error string that gives more information about the results of the request. The maximum length of the error string is 256 bytes. The error string is NULL terminated.</p> <p>The <code>mErr</code> structure might also contain information about the version of the plug-in implementation and the plug-in implementation specific data such as the configuration settings.</p> <p>Any strings in the <code>mErr</code> structure are encoded in IBM-1047 CCSID.</p>

Function return values

The `safMappingLookup()` function returns a number of pieces of information to help you with problem determination. The SAF return value, the plug-in return code (`mErr->mpiReturnCode`), the plug-in reason code (`mErr->mpiReasonCode`), and an error string (`mErr->message`) are returned by the plug-in interface code or the plug-in implementation. In addition, the version of the plug-in interface (`mErr->mpiVersion`) and any plug-in implementation specific data (`mErr->mpiInfo`) can be found in the `SafmapErr` structure.

Special processing occurs for parameter errors (`mErr->mpiReturnCode == SAFMAP_ERROR_PARMERR`) and unsupported credential types (`mErr->mpiReturnCode == SAFMAP_ERROR_NOTSUP`). The plug-in reason code (`mErr->mpiReasonCode`) contains a number identifying which parameter in the parameter list is in error.

The cleanup of the connection with the plug-in implementation occurs when the SAF return value is `SAFMAP_ERROR` or `SAFMAP_SEVERE`.

The SAF return values and the plug-in return codes (`mErr->mpiReturnCode`) are listed in the following table. These SAF return values and plug-in return codes are standard across the plug-in implementations. The plug-in reason code (`mErr->mpiReasonCode`) is unique to the plug-in implementation, except as previously noted.

Table 297. The SAF return values and the plug-in reason codes for the <code>safMappingLookup()</code> function		
SAF return value	Plug-in Return Code (<code>mErr->mpiReturnCode</code>)	Explanation
0	<code>SAFMAP_ONE_RETURNED (1)</code>	One of the SAF user credentials was returned.
SAFMAP_WARNING (4)	<code>SAFMAP_NONE_RETURNED (0)</code>	No user credential mapping was found, or the user portion of the credential is longer than 8 characters when the default plug-in implementation is used.
SAFMAP_WARNING (4)	<code>SAFMAP_MANY_FOUND (2)</code>	More than one set of SAF user credentials were found for the source user credentials.
SAFMAP_WARNING (4)	<code>SAFMAP_ERROR_NOTSUP (24)</code>	One of the parameters contains a credential type that is not supported. Check <code>mErr->mpiReasonCode</code> to identify the parameter that is in error.
SAFMAP_ERROR (8)	<code>SAFMAP_ERROR_NOTCONN (20)</code>	The plug-in implementation lost the connection with its data source during a credential lookup. Call the <code>safMappingInit()</code> function again to reestablish the connection.
SAFMAP_SEVERE (12)	<code>SAFMAP_ERROR_INTERFACE(16)</code>	The plug-in interface detected a problem internal to the <code>irrspim</code> dll.
SAFMAP_SEVERE (12)	<code>SAFMAP_ERROR_PARMERR (28)</code>	The parameter list for the plug-in contains an error. Check <code>mErr->mpiReasonCode</code> to identify the parameter that is in error.
SAFMAP_SEVERE (12)	<code>SAFMAP_ERROR_PLUGIN (32)</code>	An error internal to the plug-in implementation occurred.
SAFMAP_SEVERE (12)	<code>SAFMAP_ERROR_SETUP (36)</code>	The plug-in detected a problem in how the plug-in implementation is configured.

SAF user mapping plug-in termination function – `safMappingTerm()`

Function

The `safMappingTerm()` function cleans up any resources that the plug-in implementation used. This function must be called after the last call to `safMappingLookup()` to ensure that the storage is freed, network connections are broken, and DLLs are closed.

Format

```
#include <irrspim.h>

int safMappingTerm(
    SafmapHandle * mh,
    SafmapErr     * mErr
);
```

Requirements

1. Language: C or C++.
2. Authorization: Problem program or supervisor state, any key.
3. Dispatchable unit mode: Task.
4. Cross-memory mode: PASN=HASN.
5. AMODE: 31 bit.
6. RMODE: Any.
7. ASC Mode: Primary.
8. Serialization: Enabled for interrupts.
9. Locks: No locks held.
10. Control parameters: All storage must be in the primary address space.

RACF authorization

The calling application does not require special authorization before calling `safMappingTerm()` when using the default SAF user mapping plug-in implementation. However, the calling application requires authorization to create a mapping handle. See [“SAF user mapping plug-in initialization function – `safMappingInit\(\)`” on page 448](#) for more information.

Note: Other SAF user mapping plug-in implementations might have different authorization requirements.

Usage notes

The `safMappingTerm()` parameters behave as follows:

Table 298. The parameters of the <code>safMappingTerm()</code> function		
Parameter	Input/output	Description
<i>mh</i>	Input/output	(Required) A pointer to a <code>SafmapHandle</code> structure. The <code>mh</code> handle is used by the plug-in interface and implementations to anchor data that must persist across the calls to the plug-in interface.

Table 298. The parameters of the `safMappingTerm()` function (continued)

Parameter	Input/output	Description
mErr	Input/output	<p>(Required) A pointer to a <code>SafmapErr</code> structure.</p> <p>The <code>mErr</code> structure contains more detailed information about the success or failure of the request to the plug-in interface.</p> <p>The <code>mErr</code> structure contains a return code, a reason code provided by the plug-in implementation, and an error string that gives more information about the results of the request. The maximum length of the error string is 256 bytes. The error string is NULL terminated.</p> <p>The <code>mErr</code> structure might also contain information about the version of the plug-in implementation and the plug-in implementation specific data such as the configuration settings.</p> <p>Any strings in the <code>mErr</code> structure are encoded in IBM-1047 CCSID.</p>

Function return values

The `safMappingTerm()` function returns a number of pieces of information to help you with problem determination. The SAF return value, The plug-in return code (`mErr->mpiReturnCode`), the plug-in reason code (`mErr->mpiReasonCode`), and an error string (`mErr->message`) are returned by the plug-in interface code and the plug-in implementation. In addition, the version of the plug-in interface (`mErr->mpiVersion`) and any plug-in implementation specific data (`mErr->mpiInfo`) can be found in the `SafmapErr` structure.

Special processing occurs for parameter errors (`mErr->mpiReturnCode == SAFMAP_ERROR_PARMERR`) and unsupported credential types (`mErr->mpiReturnCode == SAFMAP_ERROR_NOTSUP`). The plug-in implementation reason code (`mErr->mpiReasonCode`) contains a number identifying which parameter in the parameter list is in error.

The cleanup of the connection with the plug-in implementation occurs when the SAF return value is `SAFMAP_ERROR` or `SAFMAP_SEVERE`.

The SAF return values and the plug-in return codes (`mErr->mpiReturnCode`) are listed in the following table. These SAF return values and plug-in return codes are standard across the plug-in implementations. The plug-in reason code (`mErr->mpiReasonCode`) is unique to the plug-in implementation, except as previously noted.

Table 299. The SAF return values and the plug-in reason codes for the `safMappingTerm()` function

SAF return value	Plug-in Return Code (<code>mErr->mpiReturnCode</code>)	Explanation
0	0	Success
SAFMAP_WARNING (4)	SAFMAP_ERROR_NOTSUP (24)	One of the parameters contains a credential type that is not supported. Check <code>mErr->mpiReasonCode</code> to identify the parameter that is in error.
SAFMAP_SEVERE (12)	SAFMAP_ERROR_INTERFACE(16)	The plug-in interface detected a problem internal to the <code>irrspim</code> dll.
SAFMAP_SEVERE (12)	SAFMAP_ERROR_PARMERR (28)	The parameter list for the plug-in contains an error. Check <code>mErr->mpiReasonCode</code> to identify the parameter that is in error.
SAFMAP_SEVERE (12)	SAFMAP_ERROR_PLUGIN (32)	An error internal to the plug-in implementation occurred.
SAFMAP_SEVERE (12)	SAFMAP_ERROR_SETUP (36)	The plug-in detected a problem in how the plug-in implementation is configured.

The irrspim.h header file

This topic contains a copy of the irrspim.h header file.

```

/**START OF SPECIFICATIONS*****
*
* Macro Name: irrspim
*
* Descriptive Name: SAF user mapping plug-in interface
* -----
* Proprietary statement:
*
* LICENSED MATERIALS - PROPERTY OF IBM
* THIS MACRO IS "RESTRICTED MATERIALS OF IBM"
* 5637-A01 (C) COPYRIGHT IBM CORP. 2006
*
* Status = HBB7730
* -----
*
* EXTERNAL CLASSIFICATION: PI
* END OF EXTERNAL CLASSIFICATION:
*
* Component: SC1BN
*
* Function:
* This file contains C/C++ function prototypes and
* related definitions for the SAF user mapping plug-in
* interface.
*
* Prototypes defined:
*
* Function          Service
* -----
* safMappingInit()  Initializes the connection to the plug-in
*                   implementation.
* safMappingLookup() Returns z/OS user credentials that have a
*                   mapping from the source user credentials.
* safMappingTerm()  Closes the connection with the plug-in
*                   implementation and cleans up resources.
*
*
* Method of Access:
* Add to the source program:
*   #include <irrspim.h>
*
* Include in the link edit:
*   irrspim.x
*
**END OF SPECIFICATIONS***/

```

Figure 16. The irrspim.h header file

```

#ifndef IRRSPIM_h
#define IRRSPIM_h
#ifdef __cplusplus
    #pragma info(none)
    extern "C" {
#else
    #pragma nomargins nosequence
    #pragma checkout(suspend)
#endif

/*--- Constants -----*/

/*-- SAF and mapping plug-in return codes -----*/
#define SAFMAP_OK          0
#define SAFMAP_WARNING    4
#define SAFMAP_ERROR      8
#define SAFMAP_SEVERE    12

/*-- Additional mapping plug-in return codes --*/
#define SAFMAP_ERROR_INTERFACE 16
#define SAFMAP_ERROR_NOTCONN  20
#define SAFMAP_ERROR_NOTSUP   24
#define SAFMAP_ERROR_PARMERR   28
#define SAFMAP_ERROR_PLUGIN   32
#define SAFMAP_ERROR_SETUP    36
#define SAFMAP_MANY_FOUND      2
#define SAFMAP_NONE_RETURNED   0
#define SAFMAP_ONE_RETURNED    1

/*-- credsType values -----*/
#define SAFMAP_REGISTRY_USER 0
#define SAFMAP_USER_ONLY    1

/*-- Miscellaneous -----*/
#define SAFMAP_EYECATCHER    "SAFMAPH"
#define SAFMAP_DEFAULT_CCSID 1047
#define SAFMAP_DEFAULT_PLUGIN
NULL

```

Figure 17. The *irrspim.h* header file (cont.)


```

/*--- Typedefs for parameters -----*/
typedef struct _SafmapCreds {
    int    credsType;
    int    credsCCSID;
    union {

        struct _UserOnly {
            char * user;
        } UserOnly;

        struct _RegistryUser {
            char * registry; /*-- Syntax is plug-in specific. --*/
            char * user;
        } RegistryUser;

    } credsData;
} SafmapCreds;

typedef struct _SafmapErr {
    int    mpiReturnCode;
    int    mpiReasonCode;
    char    mpiVersion[128];
    char    mpiInfo[128];
    char    message[256];
} SafmapErr;

typedef struct _SafmapHandle {
    char    eyecatcher[8];
    void * mpiData;
    void * dllHandle;
    void * rmapinit;
    void * rmaplook;
    void * rmapterm;
} SafmapHandle;

typedef struct _SafmapResult {
    int    bytesAvailable;
    int    credentialLen;
    SafmapCreds resultCreds;
} SafmapResult; /*-- SafmapCreds follow this header --
*/

```

Figure 18. The *irrspim.h* header file (cont.)

```

/*--- safMappingInit() -----*/
int safMappingInit(
    SafmapHandle * mh, /*-- (Required) mapping handle --*/
    char * dllName, /*-- (Optional) defaults to irrspime --*/
    SafmapErr * mErr /*-- (Required) mapping error --*/
);

/*--- safMappingLookup() -----*/
int safMappingLookup(
    SafmapHandle * mh, /*-- (Required) mapping handle --*/
    SafmapCreds * sUser, /*-- (Required) source user credentials --*/
    char * aData, /*-- (Optional) application data --*/
    SafmapResult * mResult, /*-- (Required) mapping result --*/
    SafmapErr * mErr /*-- (Required) mapping error --*/
);

/*--- safMappingTerm() -----*/
int safMappingTerm(
    SafmapHandle * mh, /*-- (Required) mapping handle --*/
    SafmapErr * mErr /*-- (Required) mapping error --*/
);

#ifdef __cplusplus
}
#endif

#endif /* IRRSPIM_h */

```

Figure 19. The *irrspim.h* header file (cont.)

Chapter 13. Generic name translate service (IRRGNT00)

The generic name translate service (IRRGNT00) translates a generic RACF profile name from internal format to external format.

Before calling IRRGNT00, the application must locate the address of the service. You can find this address in the RCVTGENT field of the RACF communications vector table (RCVT). To locate the address of the RCVT, which is mapped by the ICHPRCVT macro, use the CVTRAC field in the MVS communications vector table (CVT).

Function

The service is intended to be used to translate generic names that are provided to the RACROUTE REQUEST=FASTAUTH postprocessing exits, ICHRF02, and ICHRF04.

Environment

The requirements for the caller are these:

Minimum authorization:

Problem state, key 8

Dispatchable unit mode:

Task

Cross memory mode:

PASN=HASN=SASN

AMODE:

31 bit

ASC mode:

Primary

Interrupt status:

Enabled or disabled for interrupts for I/O or external interrupts

Locks:

The caller can hold locks, but is not required to hold any

Control parameters:

None

Restrictions

The service must not be invoked from within the RACROUTE REQUEST=FASTAUTH postprocessing exits, ICHRF02, and ICHRF04.

Input register information

When the service is invoked, register 1 must contain a pointer to the address of the parameter list described in [“Parameters” on page 462](#).

Output register information

When control returns to the caller, the general purpose registers (GPRs) contain the following information:

Register Contents

0	Unchanged
1	Pointer to the address of the parameter list
2-14	Unchanged
15	Return code

Parameters

A single parameter consists of a consecutive string of the following fields:

function_code

Fullword; Input:

X'00000001'

Translate profile or member name from internal format to external format.

Internal format name length

Fullword; input.

Internal format name

384 characters, padded with X'00'; input.

External format name length

Fullword, must be 256 for input; input or output.

External format name

256 characters, padded with blanks; output.

Return_code

Fullword; output.

Reason_code

Fullword; output.

Invoking the generic name translate service

Following is an example of a generalized programming technique you can use with assembler language to invoke this service. It is not intended to be syntactically correct.

L 3,CVTPTR	LOAD THE ADDRESS OF THE CVT
USING CVT,3	ESTABLISH ADDRESSABILITY TO THE CVT
L 4,CVTRAC	LOAD THE ADDRESS OF THE RCVT
USING RCVT,4	ESTABLISH ADDRESSABILITY TO THE RCVT
L 15,RCVTGENT	
CALL (15),(PARMADDR)	
...	
PARMADDR DC A(GNTLIST)	ADDRESS OF PARAMETER LIST
*	
GNTLIST DS 0F	IRRGNT00 PARAMETER LIST
FUNCODE DS F	FUNCTION CODE
INTNAML DS F	INTERNAL FORMAT NAME LENGTH (INPUT)
INTNAME DS CL384	INTERNAL FORMAT NAME (INPUT)
EXTNAML DS F	EXTERNAL FORMAT NAME LENGTH (INPUT/OUTPUT)
EXTNAME DS CL256	EXTERNAL FORMAT NAME (OUTPUT)
RETCODE DS F	RETURN CODE (OUTPUT)
RESCODE DS F	REASON CODE (OUTPUT)

Return and reason codes

IRRGNT00 returns the following return and reason codes in the parameter list:

Return code

Meaning

0

Function successful

Reason code

Meaning

0

Function successful.

4

Function not processed; parameter error

Reason code

Meaning

0

Incorrect function code

4

External format name length is not valid.

8

Internal format name length is not valid.

8

Function failed

Reason code

Meaning

0

Name could not be translated.

Note: The return code is also in Register 15 upon return to the caller.

Chapter 14. IRRUTIL: REXX interface to R_admin extract

IRRUTIL is a program that creates a set of REXX stem variables for several categories of RACF information.

1. The contents of RACF profiles. The stem variables are designed to reference profile information in one of the following ways:
 - Directly by field name, in the case where the field name is known by the caller.
 - The set of segment and field names are returned so that a caller can see which fields exist without any previous knowledge of the RACF schema.
2. System-wide settings (SETROPTS values).
3. RACF remote sharing facility (RRSF) information.
4. Class Descriptor Table (CDT) entries.

Except for the CDT function, IRRUTIL calls the R-admin callable service (IRRSEQ00) to extract RACF information that is converted into a set of REXX stem variables. Users of IRRUTIL should have some familiarity with R_admin extract functions. However, you need not understand the input and output parameters of the R_admin extract functions before coding to the IRRUTIL interface.

IRRUTIL is invoked by REXX in problem state.

Some aspects of the interface that are determined by R_admin are not documented in this topic. These include, but are not limited to, the following:

- Authorization requirements
- Segment and field names

For more information about authorization requirements, see *z/OS Security Server RACF Callable Services* and the 'Authorization required' section for the relevant RACF command, in *z/OS Security Server RACF Command Language Reference*.

Tip: Find helpful samples of IRRUTIL programming on the [RACF home page \(www.ibm.com/products/resource-access-control-facility/resources\)](http://www.ibm.com/products/resource-access-control-facility/resources).

Parameters

You can invoke IRRUTIL using the following REXX statement:

```
/* REXX */
myrc=IRRUTIL(command,type,profile,stem,prefix,generic)
```

The following table defines the IRRUTIL parameters:

Table 300. IRRUTIL parameters

Parameter	Definition
command	<ul style="list-style-type: none"> 'EXTRACT' to extract the contents of the entity specified with the type and profile parameters. 'EXTRACTN' to extract the contents of the entity that follows the specified entity alphabetically. This function can be used iteratively to return all profiles in a class, or all CDT entries, by specifying the output profile name (and, for RACF profiles, the generic indicator) from the previous iteration as input to the subsequent iteration. <p>Notes:</p> <ol style="list-style-type: none"> When iteratively retrieving general resource profiles starting at the beginning (specifying profile as a single blank), the discrete profiles are returned in alphabetic order, followed by the generic profiles in alphabetic order. R_admin automatically transitions from discrete profiles to generic profiles. When iteratively retrieving data set profiles starting at the beginning (specifying profile as a single blank), for each high level qualifier, generic profiles are returned in alphabetic order, followed by discrete profiles in alphabetic order.
type	<p>The uppercase class name from which to extract the specified profile. USER, GROUP, CONNECT, DATASET, _SETOPTS, _RRSFEXTR, _CDT and all general resource classes are supported.</p> <p>Notes:</p> <ol style="list-style-type: none"> A user-to-group connection can be extracted by specifying the CONNECT class and providing a profile name using the form <i>userID.group-name</i>. SETOPTS settings can be obtained by specifying "_SETOPTS" as both the class name and the profile name. RACF subsystem and remote sharing information can be obtained by specifying "_RRSFEXTR" as both the class name and the profile name. A RACF CDT entry (static or dynamic) can be obtained by specifying "_CDT" as the class name and the upper-case name of the class as the profile name.
profile	<p>The profile to extract, or to be used as the basis for an EXTRACTN request.</p> <p>Note: The profile name is case-sensitive. See the type parameter description for more information.</p>
stem	<p>The REXX stem variable name to contain returned data. The stem must not exceed 32 characters or include a trailing period ("."). IRRUTIL initializes the stem variable to null before setting fields unless the stem contains a period (".").</p> <p>Note: Specifying a stem that contains a period requires some familiarity with REXX programming techniques. For details, see “Specifying a period in the stem name” on page 472.</p>
prefix	<p>An optional prefix to add to all stem variable name parts other than the actual stem and the value of the variable, except when the value is a segment or field name (for example, the list of segments returned for a profile, or the list of fields associated with a segment). The prefix reduces collisions with other REXX variables that might be defined within the calling program. The prefix must not exceed 32 characters and must not contain a period.</p>

Table 300. IRRXUTIL parameters (continued)

Parameter	Definition			
generic	An optional indicator for use with general resources and DATASET profiles. The value must be uppercase “TRUE”, “FALSE”, or “MATCH“. The default value for generic is "FALSE". Depending on which value and which command is specified, generic returns different profiles:			
	For general resources:			
		Behavior		
	Command	TRUE	MATCH	FALSE
	EXTRACT	Returns the profile that covers the input profile name if there is no exact match. Note: The class must be active, and SETROPTS GENERIC must be in effect for the class, in order for a generic match to be found.	Same as True.	Returns R_admin “profile not found” return code combination when there is no exact match.
EXTRACTN	Returns the next alphabetic generic profile in the class about the input name (regardless of whether the input profile name is generic or discrete).	Not allowed.	<ul style="list-style-type: none">• If the input profile name is discrete, return the next alphabetic discrete profile if more discrete profiles exist, or the first alphabetic generic profile.• If the input profile name is generic, return the next alphabetic generic profile.	

Table 300. IRRUTIL parameters (continued)

Parameter	Definition			
generic (cont.)	For data sets:			
	Command	Behavior		
		TRUE	MATCH	FALSE
	EXTRACT	Return the generic profile that exactly matches the input profile name.	Returns the profile that covers the input profile name if there is no exact match. Set the <i>stem.VOLUME</i> variable to the desired volume prior to the IRRUTIL call. If a volume is not specified, the data set must be cataloged.	Return the discrete profile that exactly matches the input profile name. If the <i>stem.VOLUME</i> variable does not contain a volume name, and multiple discrete profiles exist for the specified name, RACF will return the first one, along with return codes 0/0/0/0/4 to indicate that additional profiles exist. To get the additional profiles, the caller can continue issuing EXTRACTN requests until the returned profile name is different.
	EXTRACTN	Indicates whether the input profile is generic or not.	Not allowed.	Indicates whether the input profile is generic or not. This should be set to the value of <i>stem.GENERIC</i> that was returned by the previous EXTRACTN request.
	Note: This parameter is ignored if type is not DATASET or a general resource class. See z/OS Security Server RACF Callable Services for additional information.			
myrc	A string of 5 blank-separated return code values. See “Return codes” on page 469 for more information.			

In addition, there are several variables that can be used to alter the behavior of IRRUTIL. These must be set by the application prior to the call to IRRUTIL. The only values that can be assigned to these variables are “0” and “1”. If the value is “1”, the requested behavior is performed.

Variables that do not apply to the specified **type** will be ignored (though an invalid value will result in an error). If a **prefix** is specified in the IRRUTIL call, then it must also be used with the names that follow. Otherwise, IRRUTIL uses a prefix value of “IRRUTIL_OPTION_” when locating the variable.

Table 301. Variables to alter the behavior of IRRXUTIL

Parameter	Definition
<prefix>NAMEONLY	Requests the profile name, but not contents, from R_admin (sets the ADMN_PROF_NAMEONLY bit)
<prefix>BASEONLY	Requests only fields in the base segment from R_admin (sets the ADMN_PROF_BASEONLY bit)
<prefix>UPPERCASE	Requests R_admin to automatically uppercase the input profile name for general resource classes that do not support mixed case names (sets the ADMN_PROF_UPPERCASE bit)
<prefix>CLASS_SETTINGS	For CDT-extract, requests the current SETROPTS options for the class. This requires authorization to perform R_admin SETROPTS extract.

Examples

1. In the following example "EXTRACT" is the command, "UNIXPRIV" is the type, "SHARED.IDS" is the profile, "MYSTEM" is the stem, "R_" is the prefix, and "FALSE" is the optional indicator for the "EXTRACT" command:

```
myrc=IRRXUTIL("EXTRACT","UNIXPRIV","SHARED.IDS","MYSTEM","R_","FALSE")
```

2. The following example is similar to the previous example but with a stem value of "RACF_USERS" and no prefix:

```
myrc=IRRXUTIL("EXTRACT","UNIXPRIV","SHARED.IDS","RACF_USERS","","FALSE")
```

3. You can use EXTRACTN to return the next profile alphabetically in a class. In the following example, the profile after "SHARED.IDS" is returned (or an R_admin return code combination of 4/4/4 if no more profiles exist):

```
myrc=IRRXUTIL("EXTRACTN","UNIXPRIV","SHARED.IDS","MYSTEM","R_","FALSE")
```

4. EXTRACTN can be used to return the first profile in a class by specifying a blank for the profile name:

```
myrc=IRRXUTIL("EXTRACTN","UNIXPRIV"," ","MYSTEM","R_","FALSE")
```

The returned profile can then be used as input on a subsequent EXTRACTN request. This technique can be used iteratively to obtain all profiles in the UNIXPRIV class.

5. Return only the BASE segment of a FACILITY class profile while setting a prefix value of "R_":

```
R_BASEONLY = "1"
```

```
myrc=IRRXUTIL("EXTRACT","FACILITY"," BPX.SUPERUSER","MYSTEM","R_","FALSE")
```

Return codes

The following table defines the return codes returned by IRRXUTIL:

Table 302. Return codes					
Description	Return code 1	Return code 2	Return code 3	Return code 4	Return code 5
Success	0	0	0	0	RACF reason code

Table 302. Return codes (continued)

Description	Return code 1	Return code 2	Return code 3	Return code 4	Return code 5
Success with warning when stem contains a period. See “Specifying a period in the stem name” on page 472 for details.	2	0	0	0	RACF reason code
Incorrect number of parameters specified.	4	Number of parameters specified.	Minimum number of parameters allowed.	Maximum number of parameters allowed.	0
Parameter error.	8	Index of incorrect parameter.	1 Incorrect length (includes missing required parameters). 2 Incorrect value. 3 Incompatible (for example, EXTRACTN with CONNECT class).	0	0
The <prefix>UPPERCASE variable contains a value other than 0 or 1	8	101	2	0	0
The <prefix>BASE ONLY variable contains a value other than 0 or 1	8	102	2	0	0
The <prefix>NAME ONLY variable contains a value other than 0 or 1	8	103	2	0	0
The <prefix>CLASS_SETTINGS variable contains a value other than 0 or 1	8	104	2	0	0

Table 302. Return codes (continued)					
Description	Return code 1	Return code 2	Return code 3	Return code 4	Return code 5
The <i>stem.VOLUME</i> variable contains a value longer than six characters	8	105	2	0	0
The <i>stem.DATASET_INDEX</i> variable contains a number longer than eight characters, or contains non-numeric characters.	8	106	2	0	0
R_admin failure.	12	12	SAF return code.	RACF return code.	RACF reason code.
For CDT extract: class not found. For CDT extract-next, no more classes	13	13	4	4	4
CDT extract error	13	13	SAF rc from RACROUTE REQUEST=STAT	RACF rc from RACROUTE REQUEST=STAT	RACF reason from RACROUTE REQUEST=STAT
Environmental error.	16	0 REXX environmental error. 4 R_admin environmental error.	Offset within IRRXUTIL where the error was encountered.	Any nonzero value is additional diagnostic information for IBM service.	0

Notes:

- When an R_admin general resource profile extract failure occurs due to a ghost generic profile (a return code string of '12 12 8 4 20'), the profile name is returned as expected, but no other profile variables are set.
- Some common IRRXUTIL return codes are:
 - 12 12 4 4 4 = profile not found
 - 12 12 8 8 24 = R_admin authorization failure

- Possible cause could be no FACILITY authorization. See [z/OS Security Server RACF Callable Services](#) for more information.

SETROPTS data

SETROPTS data is returned by R_admin in a different format than profile data. IRRUTIL converts the SETROPTS extract data into the same output format as the profile data with the following considerations:

- For profile data, repeating fields are described by a header field as documented in [R_admin reference information](#) in [z/OS Security Server RACF Callable Services](#). For SETROPTS extract, there are no header fields that are documented for repeating data. Therefore, IRRUTIL simulates a repeat field header by appending “_CT” to the name of the field as documented for R_admin. For example, the CLASSACT field returns a list of active classes. IRRUTIL returns a header field named CLASSACT_CT, followed by an instance of the CLASSACT field for each active class. The following list shows the repeat field header names returned for SETROPTS:
 - CLASSACT_CT
 - CLASSTAT_CT
 - GENCMD_CT
 - GENERIC_CT
 - GENLIST_CT
 - GLOBAL_CT
 - RACLIST_CT
 - AUDIT_CT
 - LOGALWYS_CT
 - LOGNEVER_CT
 - LOGSUCC_CT
 - LOGFAIL_CT
 - LOGDEFLT_CT
- While there is a flag field for the segment entry returned by R_admin for SETROPTS extract, it does not have the same meaning or contents as the segment descriptor flag field for profile extract. IRRUTIL always returns a value of "00000000" for *stem.segname.FLAGS* for SETROPTS data.

Specifying a period in the stem name

IRRUTIL typically changes the supplied stem variable to "" before setting the variables in the stem with RACF profile information. The following REXX statement is equivalent to changing the variable to "":

```
stem.=""
```

To use a partially filled stem variable with IRRUTIL, for example to fill in a stem with data from multiple RACF profiles, you can set a stem that contains a period (".") and make calls to fill in profile data as appropriate. In the following REXX statement, where *number* is the nth profile in a class, IRRUTIL does not clear the stem:

```
stem.number
```

This means that data from a previous call to IRRUTIL remains in the stem. It also means that the REXX default of *variable_name* is returned for any variables in the stem that are not set by IRRUTIL. Therefore, it is important that REXX applications that specify a period (".") in the stem name clear the stem before calling IRRUTIL for the first time and pass in stem names that do not conflict across multiple calls. IRRUTIL does not clear the stem when its name contains a period (".").

When a stem contains a period ("."), IRRUTIL returns a value of "2" as the main return code to highlight this warning. Return code 2 is considered a successful return code but does require that the application

perform additional steps before calling IRRXUTIL to ensure that the stem is cleared and that the stem names from one call to IRRXUTIL do not conflict with those of another call.

Examples

1. In this example, the stem MYUSER does not contain a period ("."). IRRXUTIL clears the stem of all defaults and previous values. Return code 1 is set to 0:

```
myrc=IRRXUTIL("EXTRACT","USER","FRED","MYUSER","R_","FALSE")
myrc=IRRXUTIL("EXTRACT","USER","BOB","MYUSER","R_","FALSE")
```

Therefore, after the second call the stem only contains information for "BOB".

2. In this example USERS.1 contains data about FRED. USERS.2 contains information about BOB. They do not conflict because the numbers "1" and "2" keep the data separate. IRRXUTIL does not set the stem to "". The application is responsible for setting the stem to "" before calling IRRXUTIL for the first time. The application separates the results using a stem suffix of ".1" and ".2". A class can be read into a single stem using the EXTRACTN function. Return code 1 is set to 2:

```
/* REXX */
users=""
myrc=IRRXUTIL("EXTRACT","USER","FRED","USERS.1","R_","FALSE")
myrc=IRRXUTIL("EXTRACT","USER","BOB","USERS.2","R_","FALSE")
```

3. This example returns bad result data because the stem is not initially cleared, and the same stem variable USERS.MYUSER is used. Fields that exist for FRED that do not exist for BOB, exist in the USERS.MYUSER stem after BOB is extracted. BOB's data appears in USERS.MYUSER, but data from FRED might also appear. For example, if user ID FRED has a TSO segment, but BOB does not, USERS.MYUSER contains leftover TSO data (from FRED) when BOB is extracted. Return code 1 is set to 2:

```
/* REXX */
myrc=IRRXUTIL("EXTRACT","USER","FRED","USERS.MYUSER","R_","FALSE")
myrc=IRRXUTIL("EXTRACT","USER","BOB","USERS.MYUSER","R_","FALSE")
```

REXX stem variables created by IRRXUTIL for profile and SETROPTS information

IRRXUTIL creates the REXX stem variables that are defined in the following table:

Note: The field names and properties are defined in [R_admin reference information](#) in *z/OS Security Server RACF Callable Services*. When there is only one subfield, the repeat field headers are identified in the "Field Name" column of the field tables. For example, the USER profile BASE segment WHENDAYS field indicates in parentheses that WHENDYCT is the repeat field header. For multidimensional fields such as the list of group connections for a user (the CONNECTS field in the BASE segment), the "ADDUSER/ALTUSER keyword reference" column indicates how many of the subsequent fields are subfields.

Table 303. REXX stem variables		
Variable	Description	Example
Note: The REXX stem variable examples in this table are in the USER class with a specified stem of "PROF" and without the optional prefix.		
Profile level variables		
stem.CLASS	Class name, padded with blanks to 8 characters.	PROF.CLASS = "USER"
stem.PROFILE	Profile name.	PROF.PROFILE="IBMUSER"
stem.VERSION	Version of IRRXUTIL output.	PROF.VERSION=0

Table 303. REXX stem variables (continued)

Variable	Description	Example
<i>stem</i> .GENERIC	Indicator (“TRUE” or “FALSE”) whether the returned profile is generic or not generic.	PROF.GENERIC=“FALSE”
<i>stem</i> .0	Number of segments that are returned for the profile.	PROF.0=3
<i>stem</i> . <i>n</i>	Name of <i>n</i> th segment number.	PROF.2=“OMVS”
Segment level variables		
<i>stem.segname</i> .FLAGS	Segment flags that are returned by R_admin expressed as a fullword hexadecimal string.	PROF.OMVS.FLAGS=“00000000”
<i>stem.segname</i> .0	Total number of fields that are returned for segment <i>segname</i> .	PROF.OMVS.0=4
<i>stem.segname</i> . <i>n</i>	Name of the <i>n</i> th field that is returned for segment <i>segname</i> .	PROF.OMVS.3=“PROGRAM”
Field level variables		
<i>stem.segname.fieldname</i> .OUTPUTONLY	Indicator (“TRUE” or “FALSE”) whether the field is an output-only field.	<ul style="list-style-type: none"> PROF.BASE.CREATDAT. OUTPUTONLY=“TRUE” PROF.BASE.SPECIAL. OUTPUTONLY=“FALSE”
<i>stem.segname.fieldname</i> .BOOLEAN	Indicator (“TRUE” or “FALSE”) whether the field is a Boolean field.	<ul style="list-style-type: none"> PROF.BASE.SPECIAL. BOOLEAN=“TRUE” PROF.BASE.NAME. BOOLEAN=“FALSE”
<i>stem.segname.fieldname</i> .REPEATING	Indicator (“TRUE” or “FALSE”) whether the field is a subfield of a repeat group.	<ul style="list-style-type: none"> PROF.BASE.UAUDIT. REPEATING=“FALSE” PROF.BASE.CGROU. REPEATING=“TRUE”
<i>stem.segname.fieldname</i> .0	Number of values for this field.	<ul style="list-style-type: none"> PROF.BASE.OWNER.0=1 PROF.BASE.SPECIAL.0=1
	<ul style="list-style-type: none"> For normal fields this is always 1. For repeating fields this is the number of occurrences of the field. (Same as <i>stem.segname.fieldname</i>.REPEATCOUNT in the corresponding repeat field header.) 	PROF.BASE.CGROU.0=3 (the user is connected to 3 groups)
	<ul style="list-style-type: none"> For a repeat field header, this is always 0. 	PROF.BASE.CONNECTS.0=0
<i>stem.segname.fieldname</i> . <i>n</i>	The <i>n</i> th value of the field. For normal fields, <i>n</i> is always 1.	<ul style="list-style-type: none"> PROF.BASE.OWNER.1=“IBMUSER” PROF.BASE.SPECIAL.1=“TRUE” PROF.BASE.CGROU.3=“PAYROLL” (third group name)
The following fields are only relevant for repeat group header fields:		
<i>stem.segname.fieldname</i> .REPEATCOUNT	Number of occurrences:	PROF.BASE.CONNECTS. REPEATCOUNT=5 (the user is connected to 5 groups)
	<ul style="list-style-type: none"> A nonzero value indicates a repeat field header. Any field, which is not a repeat field header, has a value of 0 for this variable. 	<ul style="list-style-type: none"> PROF.BASE.SPECIAL. REPEATCOUNT=0 PROF.BASE.CGROU. REPEATCOUNT=0

Table 303. REXX stem variables (continued)		
Variable	Description	Example
<i>stem.segname.fieldname.SUBFIELD.0</i>	<ul style="list-style-type: none"> The number of subfields that comprise the repeat group. 	<ul style="list-style-type: none"> PROF.BASE.CONNECTS. SUBFIELD.0=15 PROF.BASE.CLCNT.SUBFIELD.0=1
	<ul style="list-style-type: none"> The value is 0 for any field that is not a repeat field header. 	<ul style="list-style-type: none"> PROF.BASE.NAME.SUBFIELD.0=0 PROF.BASE.CGROU.P.SUBFIELD.0=0
<i>stem.segname.fieldname.SUBFIELD.n</i>	<p>Name of <i>n</i>th subfield of this repeat group.</p> <p>Note: This field allows an application to know which fields go together as subfields of a repeat field. The field names are also specified in the <i>stem.segname.0</i> list.</p>	<ul style="list-style-type: none"> PROF.BASE.CONNECTS. SUBFIELD.1="CGROUP" PROF.BASE.CONNECTS. SUBFIELD.15="CRESUME" PROF.BASE.CLCNT. SUBFIELD.1="CLAUTH"
The following variables are only relevant for a profile in the DATASET class:		
<i>stem.VOLUME</i>	<p>For a discrete DATASET profile, the volume on which data set resides.</p> <p>On input: Binary zeros or blanks means not specified. If specified, the field must be padded with blanks if less than 6 characters.</p> <p>On output: RACF sets this variable to contain the volume of the discrete profile returned. For a multi-volume data set, this field contains the first volume in the volume list. In all cases, the complete volume list is returned as a set of BASE segment variables.</p> <p>The caller is responsible for clearing this variable if it will interfere with subsequent application processing (e.g. if a differently named discrete profile is to be returned).</p>	<p>PROF.VOLUME=MYVOL1</p> <p>For the complete set: PROF.BASE.VOLSER.0=3 PROF.BASE.VOLSER.1=MYVOL1 PROF.BASE.VOLSER.1=MYVOL2 PROF.BASE.VOLSER.1=MYVOL3</p>
<i>stem.DATASET_INDEX</i>	<p>On output, when a volume was not specified and the returned name is part of a set of discrete profiles with the same name (differing by volume), this variable contains an index to the next profile of the set to be returned on the next EXTRACTN request. The caller should not modify this variable while extracting the set. The caller is responsible for clearing this variable if it will interfere with subsequent application processing (e.g. if a different set of profiles is to be returned).</p>	PROF.DATASET_INDEX=2

Example

The following example shows a simple program that extracts a user profile and displays the values of some fields that are defined to that profile. The example includes use of both list and non-list fields, shows how to detect the presence of a field, and how to iterate through all fields in a segment.

```
/* REXX */ /* Extract user USER01 with a stem of RACF and prefix R_ */
myrc=IRRUTIL('EXTRACT','USER','USER01','RACF','R_','FALSE')
/* Display user id */
say "The profile name is:" RACF.R_PROFILE
```

```

/* Display owner */
say "The profile owner is:" RACF.R_BASE.R_OWNER.1
/* Connection groups */
say "Connected groups:"
do group=1 to RACF.R_BASE.R_CGROUP.0
    say "    Connected to group "||RACF.R_BASE.R_CGROUP.group
end
/* Display all OMVS fields which are defined for this user */
if(RACF.R_OMVS.0<>'') then do i=1 to RACF.R_OMVS.0
    fieldName=RACF.R_OMVS.i
    say "OMVS segment field "i" "RACF.R_OMVS.i
    say "    value="RACF.R_OMVS.fieldName.1
end
/* Is R_BASE.R_SPECIAL a boolean field? */
say 'Is SPECIAL a Boolean field:' RACF.R_BASE.R_SPECIAL.R_BOOLEAN
/* Is this user RACF SPECIAL? */
say 'Is user SPECIAL:' RACF.R_BASE.R_SPECIAL.1
/* Check for the presence of the WORKATTR WAACNT field */
/* If it is set, print it, otherwise display a message */
if(RACF.R_WORKATTR.R_WAACCNT.0>0) then do
    say "WORKATTR WAACNT = "RACF.R_WORKATTR.R_WAACCNT.1
end
else do
    say "WORKATTR WAACNT is not set"
end

```

Depending on the definition of the user profile, the result of the previous example looks similar to the following output:

```

The profile name is: USER01
The profile owner is: IBMUSER
Connected groups:
    Connected to group G1
    Connected to group G2
    Connected to group G3
    Connected to group SYS1
OMVS segment field 1 R_UID
value=6543
Is SPECIAL a Boolean field: TRUE
Is user SPECIAL: FALSE
WORKATTR WAACNT is not set

```

REXX stem variables created by IRRXUTIL for RACF subsystem and remote sharing information

IRRXUTIL creates the REXX stem variables representing RRSF settings and target definitions. The data is returned in mixed case. The variables are defined in the following table:

Table 304. RRSF extract stem variables		
Variable	Description	Example
Note: The REXX stem examples in the last column of this table use a specified stem of "RACF" and without the optional prefix.		
RRSF global settings		
The RRSF global settings are always returned when the caller is authorized to the IRR.RADMIN.EXTRACT.RRSF resource in the FACILITY class. This occurs whether the caller has OPERCMDS authority to the TARGET LIST and SET LIST functions or not.		
<i>stem</i> .PREFIX_CHAR	Prefix character. One through 8 characters.	RACF.PREFIX_CHAR=<
<i>stem</i> .SUBSYS_NAME	RACF subsystem name. One through 4 characters.	RACF.SUBSYS_NAME=RSUB
<i>stem</i> .SUBSYS_USER	RACF subsystem user ID. One through 8 characters.	RACF.SUBSYS_NAME=RACFSUB
<i>stem</i> .SUBSYS_TRUSTED	Is the RACF subsystem running with the TRUSTED attribute? Value is 1 or 0.	RACF.SUBSYS_TRUSTED=1

Table 304. RRSF extract stem variables (continued)		
Variable	Description	Example
<i>stem</i> .SUBSYS_PRIVILEGED	Is the RACF subsystem running with the PRIVILEGED attribute? Value is 1 or 0.	RACF.SUBSYS_PRIVILEGED=0
<i>stem</i> .NODE_DATA_PRESENT	Allowable values are 0 and 1. If 0, the variables containing RRSF node information are not set. The caller does not have OPERCMDS authority to the TARGET LIST function.	RACF.NODE_DATA_PRESENT=0
<i>stem</i> .SET_DATA_PRESENT	Allowable values are 0 and 1. Allowable values are 0 and 1. If 0, the variables containing SET data are not set. The caller does not have OPERCMDS authority to the SET LIST function.	RACF.SET_DATA_PRESENT=1
The following variables are set when <i>stem</i> .SET_DATA_PRESENT=1.		
<i>stem</i> .TRACE_SSL	Indicates whether tracing of SSL requests is active or not.	RACF.TRACE_SSL=0
<i>stem</i> .TRACE_APPC	Indicates whether tracing of APPC requests is active or not.	RACF.TRACE_APPC=0
<i>stem</i> .TRACE_RRSF	Indicates whether tracing of RRSF requests is active or not.	RACF.TRACE_RRSF=1
<i>stem</i> .TRACE_IMAGE	Indicates whether tracing of command images requests is active or not.	RACF.TRACE_IMAGE=0
<i>stem</i> .FULLRRSF COMM	Is SET FULLRRSF COMM enabled on this system? Value is 1 or 0.	RACF.FULLRRSF COMM=1
<i>stem</i> .ADAU	Is the Automatic Direction of Application Updates function active or not? Value is 1 or 0.	RACF.ADAU=1
<i>stem</i> .ADAU.NOTIFY.0	Number of IDs for notification of the ADAU function success or failure. <ul style="list-style-type: none"> Value is between 0 and 4, inclusive. If 0, there are no notification user IDs defined. 	RACF.ADAU.NOTIFY.0=2
<i>stem</i> .ADAU.NOTIFY.#.ID	ID to notify.	RACF.ADAU.NOTIFY.1.ID=BIGJOE
<i>stem</i> .ADAU.NOTIFY.#.NODE	NODE for notification.	RACF.ADAU.NOTIFY.1.NODE=NODE1
<i>stem</i> .ADAU.NOTIFY.#.LEVEL	Level of success at which notification occurs. Level is WARN, FAIL, or ALWAYS.	RACF.ADAU.NOTIFY.1.LEVEL=FAIL
<i>stem</i> .ADAU.OUTPUT.0	Number of IDs that are defined for output of the ADAU function. Value is between 0 and 4, inclusive.	RACF.ADAU.OUTPUT.0=2
<i>stem</i> .ADAU.OUTPUT.#.ID	ID for password change output.	RACF.ADAU.OUTPUT.1.ID=BIGJOE

Table 304. RRSF extract stem variables (continued)

Variable	Description	Example
<i>stem</i> .ADAU.OUTPUT.#.NODE	NODE for password change output.	RACF.ADAU.OUTPUT.1.NODE=NODE1
<i>stem</i> .ADAU.OUTPUT.#.LEVEL	Level of success for output to occur. Level is WARN, FAIL, or ALWAYS.	RACF.ADAU.OUTPUT.1.LEVEL=FAIL
<i>stem</i> .AUTODIRECT	Is the Automatic Command Direction function active or not? Value is 1 or 0.	RACF.AUTODIRECT=1 or 0
<i>stem</i> .AUTODIRECT.NOTIFY.0	Number of Ids for notification of the AUTODIRECT function success or failure. Value is between 0 and 4, inclusive.	RACF.AUTODIRECT.notify.0=2
<i>stem</i> .AUTODIRECT.NOTIFY.#.ID	ID to notify.	RACF.AUTODIRECT.NOTIFY.1.ID=BIGJOE
<i>stem</i> .AUTODIRECT.NOTIFY.#.NODE	NODE for notification.	RACF.AUTODIRECT.NOTIFY.1.NODE=NODE1
<i>stem</i> .AUTODIRECT.NOTIFY.#.LEVEL	Level of success at which notification occurs. Level is WARN, FAIL, or ALWAYS.	RACF.AUTODIRECT.NOTIFY.1.LEVEL=FAIL
<i>stem</i> .AUTODIRECT.OUTPUT.0	Number of Ids that are defined for output of the AUTODIRECT function. Value is between 0 and 4, inclusive.	RACF.AUTODIRECT.OUTPUT.0=2
<i>stem</i> .AUTODIRECT.OUTPUT.#.ID	ID for password change output.	RACF.AUTODIRECT.OUTPUT.1.ID=BIGJOE
<i>stem</i> .AUTODIRECT.OUTPUT.#.NODE	NODE for password change output.	RACF.AUTODIRECT.OUTPUT.1.NODE=NODE1
<i>stem</i> .AUTODIRECT.OUTPUT.#.LEVEL	Level of success for output to occur. Level is WARN, FAIL, or ALWAYS.	RACF.AUTODIRECT.OUTPUT.1.LEVEL=FAIL
<i>stem</i> .APD	Is the Automatic Password Direction function active or not? Value is 1 or 0.	RACF.APD=1 or 0
<i>stem</i> .APD.NOTIFY.0	Number of Ids for notification of the APD function success or failure. Value is between 0 and 4, inclusive.	RACF.APD.NOTIFY.0=2
<i>stem</i> .APD.NOTIFY.#.ID	ID to notify.	RACF.APD.NOTIFY.1.ID=BIGJOE
<i>stem</i> .APD.NOTIFY.#.NODE	NODE for notification.	RACF.APD.NOTIFY.1.NODE=NODE1
<i>stem</i> .APD.NOTIFY.#.LEVEL	Level of success at which notification occurs. Level is WARN, FAIL, or ALWAYS.	RACF.APD.NOTIFY.1.LEVEL=FAIL
<i>stem</i> .APD.OUTPUT.0	Number of Ids that are defined for output of the APD function. Value is between 0 and 4, inclusive.	RACF.APD.OUTPUT.0=2
<i>stem</i> .APD.OUTPUT.#.ID	ID for password change output.	RACF.APD.OUTPUT.1.ID=BIGJOE

Table 304. RRSF extract stem variables (continued)		
Variable	Description	Example
<i>stem</i> .APD.OUTPUT.#.NODE	NODE for password change output.	RACF.APD.OUTPUT.1.NODE=NODE1
<i>stem</i> .APD.OUTPUT.#.LEVEL	Level of success for output to occur. Level is WARN, FAIL, or ALWAYS.	RACF.APD.OUTPUT.1.LEVEL=FAIL
<i>stem</i> .PWSYNC	Is the Password Synchronization function active or not. Value is 1 or 0.	RACF.PWSNC=1 or 0
<i>stem</i> .PWSYNC.NOTIFY.0	Number of Ids for notification of the PWSYNC function success or failure. Value is between 0 and 4, inclusive.	RACF.PWSYNC.NOTIFY.0=2
<i>stem</i> .PWSYNC.NOTIFY.#.ID	ID to notify.	RACF.PWSYNC.NOTIFY.1.ID=BIGJOE
<i>stem</i> .PWSYNC.NOTIFY.#.NODE	NODE for notification.	RACF.PWSYNC.NOTIFY.1.NODE=NODE1
<i>stem</i> .PWSYNC.NOTIFY.#.LEVEL	Level of success at which notification occurs. Level is WARN, FAIL, or ALWAYS.	RACF.PWSYNC.NOTIFY.1.LEVEL=FAIL
<i>stem</i> .PWSYNC.OUTPUT.0	Number of Ids that are defined for output of the PWSYNC function. Value is between 0 and 4, inclusive.	RACF.PWSYNC.OUTPUT.0=2
<i>stem</i> .PWSYNC.OUTPUT.#.ID	ID for password change output.	RACF.PWSYNC.OUTPUT.1.ID=BIGJOE
<i>stem</i> .PWSYNC.OUTPUT.#.NODE	NODE for password change output.	RACF.PWSYNC.OUTPUT.1.NODE=NODE1
<i>stem</i> .PWSYNC.OUTPUT.#.LEVEL	Level of success for output to occur. Level is WARN, FAIL, or ALWAYS.	RACF.PWSYNC.OUTPUT.1.LEVEL=FAIL
RRSF node definition information. The following variables are set when <i>stem</i> .NODE_DATA_PRESENT=1.		
<i>stem</i> _NODE_DATA_TRUNCATED	Allowable values are 0 and 1. If 0, there are more nodes that are defined to RRSF than were returned. This is an unlikely occurrence. It happens when there are too many RRSF nodes defined and there is not enough memory to return them all.	RACF.NODE_DATA_TRUNCATED=0
<i>stem</i> .0	Number of nodes that are defined to RRSF. The '#' in subsequent variables must be smaller than or equal to this value.	RACF.0=4
<i>stem</i> .#.NODE	Node name.	RACF.1.NODE=NODEA
<i>stem</i> .#.SYSNAME	System name within MSN (if any).	RACF.1.SYSNAME=SYS1
<i>stem</i> .#.MSN	Value is 1 if this node is part of an MSN, otherwise, the value is 0.	RACF.1.MSN=1
<i>stem</i> .#.LOCAL	Value is 1 if this is the local node.	RACF.1.LOCAL=0

Table 304. RRSF extract stem variables (continued)

Variable	Description	Example
<i>stem.#</i> .MSN_MAIN	Value is 1 if this system is the MAIN system on an MSN.	RACF.1.MSN_MAIN=0
<i>stem.#</i> .PREFIX	Workspace data set prefix, as specified on TARGET command.	RACF.1.PREFIX=RRSF
<i>stem.#</i> .WDSQUAL	Workspace data set qualifier, as specified on TARGET command.	RACF.2.WDSQUAL=QUAL
<i>stem.#</i> .VOLUME	Workspace data set volume, as specified on TARGET command.	RACF.1.VOLUME=TEMP01
<i>stem.#</i> .FILESIZE	Workspace data set file size, as specified on TARGET command.	RACF.1.FILESIZ=1000
<i>stem.#</i> .DATACLAS	Workspace data set data class name, as specified on TARGET command.	RACF.2.DATACLAS=DCLAS1
<i>stem.#</i> .STORCLAS	Workspace data set storage class name, as specified on TARGET command.	RACF.2.STORCLAS=SCLAS1
<i>stem.#</i> .MGMTCLAS	Workspace data set management class name, as specified on TARGET command.	RACF.2.MGMTCLAS=MCLAS1
<i>stem.#</i> .DESCRIPTION	RRSF node description, as specified on the TARGET command.	RACF.1.DESCRPTION=FIRST NODE
<i>stem.#</i> .PARTNER_DENYING_INBOUND	If true, the partner node does not accept work from the local node. This variable is accurate as of the last successful connection to the partner. If there is no connection, the value is 0.	RACF.1.PARTNER_DENYING_INBOUND=0
<i>stem.#</i> .DENY_INBOUND	If true, the local node does not accept work from the remote node.	RACF.1.DENY_INBOUND=0
<i>stem.#</i> .DENIED_COUNT	Number indicating how many requests sent from the remote target were denied by the local node.	RACF.1.DENIED_COUNT=420
<i>stem.#</i> .PROTOCOL	For a remote node: <ul style="list-style-type: none"> (<i>stem.#</i>.LOCAL=0): TCP or APPC. For a local node: <ul style="list-style-type: none"> (<i>stem.#</i>.LOCAL=1): a blank separated list of the protocols defined. 	Remote: <ul style="list-style-type: none"> RACF1.PROTOCOL=APPC Local: <ul style="list-style-type: none"> RACF1.PROTOCOL=APPC TCP
<i>stem.#</i> .APPC_LUNAME	APPC LUNAME, as specified on the TARGET command.	RACF.1.LUNAME=MF1AP001
<i>stem.#</i> .APPC_MODENAME	APPC MODENAME, as specified on the TARGET command.	RACF.1.MODENAME="" (For example, the field has no value)
<i>stem.#</i> .APPC_NETNAME	APPC NETNAME.	RACF.1.NETNAME=""
<i>stem.#</i> .APPC_TP_NAME	APPC TP_NAME, as specified on the TARGET command.	RACF.1.TP_NAME=""

Table 304. RRSF extract stem variables (continued)		
Variable	Description	Example
<i>stem.#.APPC_LISTENER_STATUS</i>	Status of APPC listener. Status is INACTIVE, INITIALIZING, or ACTIVE. Is only set for LOCAL node. When the local node is a multisystem node, the value is only set for the local system (SYSNAME matches the z/OS system name in CVTSNAME).	RACF.1.APPC_LISTENER_STATUS:ACTIVE
<i>stem.#.TCP_PORT</i>	TCP port, as specified on TARGET command.	RACF.1.TCP_PORT=18136
<i>stem.#.TCP_ADDRESS</i>	TCP address, as specified on the TARGET command.	RACF.1.TCP_address=nodea.globo.com
<i>stem.#.TCP_LISTENER_STATUS</i>	Status of TCP listener. Status is INACTIVE, INITIALIZING, or ACTIVE. Is only set for LOCAL node. When the local node is a multisystem node, the value is only set for the local system (SYSNAME matches the z/OS system name in CVTSNAME).	RACF.1.TCP_LISTENER_STATUS:ACTIVE
<i>stem.#.TCP_IPADDRESS</i>	Resolved TCP address.	RACF.1.TCP_IPADDRESS=12.23.44.23
<i>stem.#.TCP_TLS_CERTU</i>	TCP TLS certificate user ID.	RACF.1.TCP_TLS_CERTU=IBMUSER
<i>stem.#.TCP_TLS_RULE</i>	TCP TLS rule in effect for this connection. Rule is RRSF-Client or RRSF-Server.	RACF.1.TCP_TLS_RULE=RRSF-Client
<i>stem.#.TCP_TLS_CIPHER</i>	TCP TLS cipher in effect for this connection. The cipher is one of the standard cipher suites that are defined for TLS.	RACF.1.TCP_TLS_CIPHER=35 TLS_RSA_WITH_AES_256_CBC_SHA
<i>stem.#.TCP_TLS_CLIENT_AUTH</i>	TCP TLS client author value in effect for this connection. Value is Required or SAFCheck.	RACF.1.TCP_TLS_CLIENT_AUTH=Required
<i>stem.#.STATE</i>	Connection state of this node. ???=0 Dormant Error=1 Dormant Both=2 Defined=3 Dormant Remote=4 Dormant Local=8 Operative Error=16 Operative Pending Verification=32 Operative Pending Connection=64 Operative Active=128	RACF.1.STATE=128

Table 304. RRSF extract stem variables (continued)		
Variable	Description	Example
<i>stem.#.STATE_STRING</i>	Connection state of this node in test form. One of: ??? DORMANT ERROR DORMANT BOTH DEFINED DORMANT REMOTE DORMANT LOCAL OPERATIVE ERROR OPERATIVE PENDING VERIFICATION OPERATIVE PENDING CONNECTION OPERATIVE ACTIVE	RACF.1.STATE_STRING=OPERATIVE ACTIVE
<i>stem.#.MSN_PENDING_EX_MAIN</i>	Value is 1 when this system used to be the MAIN system in an MSN, but is no longer the main, but the new main is not yet connected.	RACF.1.MSN_PENDING_EX_MAIN=0
<i>stem.#.IS_SECOND_PROTOCOL</i>	Value is 1 if this node describes a second protocol for an existing node. A second protocol node is a node that is not yet in use. The intent is to switch the communication protocol to that of the second protocol node, but this switch has not yet occurred for nodes with IS_SECOND_PROTOCOL=1.	RACF.1.IS_SECOND_PROTOCOL=0
<i>stem.#.HAS_CONVERSION_FILE</i>	Value is 1 if this node is in the middle of a protocol conversion and might convert files that are detailed in the INMSG2.xxx and OUTMSG2.xxx variables. <i>Stem.#.INMSG2.EXISTS</i> and <i>Stem.#.OUTMSG2.EXISTS</i> indicate which conversion files exist.	RACF.1.HAS_CONVERSION_FILE=0
<i>stem.#.INMSG_EXISTS</i>	Status of INMSG file. If the value is 1, then the file is allocated and with no errors. If the value is 0, the other <i>stem.#.INMSG.xxx</i> fields have no meaning.	RACF.1.INMSG_EXISTS=1
<i>stem.#.INMSG_DSNAME</i>	INMSG data set name.	RACF.1.INMSG_DSNAME=RRSF.NODE1. NODE2.INMSG
<i>stem.#.INMSG_EXTENTS</i>	Number of extents that are used by INMSG data set.	RACF.1.INMSG_EXTENTS=2
<i>stem.#.INMSG_RECORDS</i>	Number of records in INMSG data set. The number of records is dynamic and might change between the time it is read and the time it is used by a REXX program.	RACF.1.INMSG_RECORDS=23

Table 304. RRSF extract stem variables (continued)		
Variable	Description	Example
<i>stem.#.OUTMSG_EXISTS</i>	Status of OUTMSG file. If the value is 1, then the file is allocated and with no errors. If the value is 0, the other <i>stem.#.OUTMSG.xxx</i> fields have no meaning.	RACF.1.OUTMSG_EXISTS=1
<i>stem.#.MAIN_READING_OUTMSG</i>	If the value is 1, then the OUTMSG file is being read by the task that is associated with the MAIN node. Otherwise, the value is 0.	RACF.1.MAIN_READING_OUTMSG=0
<i>stem.#.OUTMSG_DSNAME</i>	OUTMSG data set name.	RACF.1.OUTMSG_DSNAME=RRSF.NODE1.NODE2.OUTMSG
<i>stem.#.OUTMSG_EXTENTS</i>	Number of extents that are used by OUTMSG data set.	RACF.1.OUTMSG_EXTENTS=2
<i>stem.#.OUTMSG_RECORDS</i>	Number of records in OUTMSG data set. The number of records is dynamic and might change between the time it is read and the time it is used by a REXX program.	RACF.1.OUTMSG_RECORDS=23
<i>stem.#.INMSG2_EXISTS</i>	Status of INMSG2 conversion file. If the value is 1, then the file is allocated and with no errors. If the value is 0, the other <i>stem.#.INMSG2.xxx</i> fields have no meaning.	RACF.1.INMSG2_EXISTS=1
<i>stem.#.INMSG2_DSNAME</i>	INMSG2 conversion data set name.	RACF.1.INMSG2_DSNAME=RRSF.NODE1.NODE2.INMSG2
<i>stem.#.INMSG2_EXTENTS</i>	Number of extents that are used by INMSG2 conversion data set.	RACF.1.INMSG2_EXTENTS=2
<i>stem.#.INMSG2_RECORDS</i>	Number of records in INMSG2 conversion data set. The number of records is dynamic and might change between the time it is read and the time it is used by a REXX program.	RACF.1.INMSG2_RECORDS=23
<i>stem.#.OUTMSG2_EXISTS</i>	Status of OUTMSG2 conversion file. If the value is 1, then the file is allocated and with no errors. If the value is 0, the other <i>stem.#.OUTMSG2.xxx</i> fields have no meaning.	RACF.1.OUTMSG2_EXISTS=1
<i>stem.#.MAIN_READING_OUTMSG2</i>	If the value is 1, then the OUTMSG2 conversion file is being read by the task that is associated with the MAIN node. Otherwise, the value is 0.	RACF.1.MAIN_READING_OUTMSG2=0
<i>stem.#.OUTMSG2_DSNAME</i>	OUTMSG2 conversion data set name.	RACF.1.OUTMSG2_DSNAME=RRSF.NODE1.NODE2.OUTMSG2

Table 304. RRSF extract stem variables (continued)

Variable	Description	Example
<i>stem.#.OUTMSG2_EXTENTS</i>	Number of extents that are used by OUTMSG2 conversion data set.	RACF.1.OUTMSG2_EXTENTS=2
<i>stem.#.OUTMSG2_RECORDS</i>	Number of records in OUTMSG2 conversion data set. The number of records is dynamic and might change between the time it is read and the time it is used by a REXX program.	RACF.1.OUTMSG2_RECORDS=23
<i>stem.#.LAST_INBOUND_DATE</i>	YYYYMMDD of last request that is received from this remote node.	RACF.1.LAST_INBOUND_DATE:20120919
<i>stem.#.LAST_INBOUND_TIME</i>	HH:MM:SS time stamp of last request that is received from this node.	RACF.1.LAST_INBOUND_TIME:08:02:24
<i>stem.#.LAST_OUTBOUND_DATE</i>	YYYYMMDD of last request that is sent to remote node.	RACF.1.LAST_OUTBOUND_DATE:20120919
<i>stem.#.LAST_OUTBOUND_TIME</i>	HH:MM:SS time stamp of last request sent to this node.	RACF.1.LAST_OUTBOUND_TIME:08:02:24
<i>stem.#.PARTNER_OS_VERSION</i>	Text representation of partner operating system version from ECVTPSEQ. When the partner is lower than z/OS V2R2, the value is 0 for this field.	RACF.1.PARTNER_OS_VERSION: 01020200 or RACF.1.PARTNER_OS_VERSION: 00000000
<i>stem.#.PARTNER_TEMPLATE_RELEASE</i>	Numeric version of partner RACF database template release. Is only set if partner node is running z/OS R13 (with APAR OA38594 applied) or higher.	RACF.1.PARTNER_TEMPLATE_RELEASE:171
<i>stem.#.PARTNER_TEMPLATE_APAR</i>	Numeric value of partner RACF database APAR level. Set only if partner is running z/OS R13 (with APAR OA38594 applied) or higher.	RACF.1.PARTNER_TEMPLATE_APAR:20
<i>stem.#.PARTNER_PARSE</i>	Partner Dynamic Parse version.	RACF.1.PARTNER_PARSE:HRF7790
<p>Each of the following variables can be used as an index to RACF.# variables (For example, variables that pertain to a specific target definition). When you know the node name (qualified by system name and protocol, where necessary), you can use it as a variable that resolves to the index of that target. Therefore, you can directly access variables that pertain to that target.</p> <p>The prefix is added to all of the variable names, if specified.</p>		
<i>LOCALNODE</i>	Index number of the LOCAL node. If using MSN, it is the LOCAL node whose SYSNAME matches the z/OS system name (CVTSNAME).	RACF.LOCALNODE=7 In this example, the local node is the seventh entry. To find the name of the local node: RACF.LOCALNODE.NAME=POKNODE1
<i>NodeName</i>	Index number of the SSN node with this name, corresponding to #. Only one of <i>NodeName</i> or <i>NodeName_sysName</i> is defined. If the node with name <i>NodeName</i> has 2 protocols that are defined, <i>NodeName</i> corresponds to the index of the node that has IS_SECOND_PROTOCOL=0.	To find the prefix of a single system node named NODE5: RACF.NODE5.PREFIX=SYS1.RRSF

Table 304. RRSF extract stem variables (continued)		
Variable	Description	Example
<i>nodeName_sysName</i>	Index number of the MSN node with this name and <i>sysname</i> corresponding to #. Either <i>nodeName</i> or <i>nodeName_sysName</i> is defined. If node <i>nodeName_sysName</i> is remote and defined with >1 protocol, <i>nodeName_sysName</i> is set to the index of the primary protocol.	To find the description of the SYS1 system in the multisystem node named NODE1: RACF.NODE1_SYS1.DESCRPTION=MAIN SYSTEM FOR TAMPA MSN
<i>nodeName_protocol</i>	Index number of this SSN node with this name, corresponding to #. Only one of <i>nodeName_protocol</i> or <i>nodeName_sysName_protocol</i> is defined.	To find the IP address of the single system node named SSN1: RACF.SSN1_TCP.TCP_IPADDRESS=1.2.3.4
<i>nodeName_sysName_protocol</i>	Index number of this MSN node with this name and <i>sysname</i> corresponding to #. Only one of <i>nodeName_protocol</i> or <i>nodeName_sysName_protocol</i> is defined.	To find the state of the APPC connection to the system named SYSA in the multisystem node named MSN5: RACF.MSN5_SYSA_APPC.STATE_STRING=OPERATIVE PENDING CONNECTION

Class descriptor entry data

Table 305. Class descriptor entry data		
Variable	Description	Example
Note: The REXX stem variable examples in this table assume a specified stem of “CLS” and no prefix.		
<i>stem.CLASSNAME</i>	Requested class name	CLS.CLASSNAME=XFACILIT
<i>stem.ID</i>	ID value for this class. The value is 0 for dynamic classes.	CLS.ID=1
<i>stem.POSIT</i>	POSIT number of class	CLS.POSIT=8
<i>stem.MAXLNTH</i>	Maximum length of a profile name	CLS.MAXLNTH=246
<i>stem.MAXLENX</i>	Maximum length of a profile name	CLS.MAXLENX=246
<i>stem.FIRSTCHAR</i>	Allowable character types in first position. This is a list of up to four types in a blank-delimited string. Possible values are ALPHA, NUMERIC, NATIONAL, and SPECIAL.	CLS.FIRSTCHAR=ALPHA NUMERIC
<i>stem.OTHERCHAR</i>	Allowable character types in remaining positions. This is a list of up to four types in a blank-delimited string. Possible values are ALPHA, NUMERIC, NATIONAL, and SPECIAL.	CLS.OTHERCHAR=ALPHA NATIONAL NUMERIC SPECIAL

Table 305. Class descriptor entry data (continued)

Variable	Description	Example
<i>stem</i> .UACC	Minimum access allowed if the access level is not set when a resource profile is defined in the class. The value is "ACEE" if the class was defined using the ICHERCDE macro, and no value was specified for the DFTUACC keyword.	CLS.UACC=NONE
<i>stem</i> .GROUPING_CLASS	Name of the associated grouping class, or null if not applicable (the requested class is not a member class)	CLS.GROUPING_CLASS=GXFACIL I
<i>stem</i> .MEMBER_CLASS	Name of the associated member class, or null if not applicable (the requested class is not a grouping class)	CLS.MEMBER_CLASS= (null: there is no associated member class)
<i>stem</i> .OPERATIONS	The OPERATIONS attribute is considered during authorization checking for this class	CLS.OPERATIONS=0
<i>stem</i> .RACLIST_ALLOWED	RACLIST allowed	CLS.RACLIST_ALLOWED=1
<i>stem</i> .RACLIST_REQUIRED	RACLIST required	CLS.RACLIST_REQUIRED=0
<i>stem</i> .GENERIC_ALLOWED	Generic profiles are allowed	CLS.GENERIC_ALLOWED=1
<i>stem</i> .GENLIST_ALLOWED	GENLIST allowed	CLS.GENLIST_ALLOWED=1
<i>stem</i> .DEFAULT_RC	Default return code	CLS.DEFAULT_RC=8
<i>stem</i> .PROFILES_ALLOWED	Profiles can be defined in this class	CLS.PROFILES_ALLOWED=1
<i>stem</i> .SECLABEL_REQUIRED	Security labels are required for profiles in this class	CLS.SECLABEL_REQUIRED=0
<i>stem</i> .EQUAL_MAC	Equal mandatory access checking is required when users attempt to access resources protected by profiles in this class	CLS.EQUAL_MAC=0
<i>stem</i> .REVERSE_MAC	Reverse mandatory access checking is required when users attempt to access resources protected by profiles in this class	CLS.REVERSE_MAC=0
<i>stem</i> .MIXED_CASE	Mixed case profile names are allowed	CLS.MIXED_CASE=0
<i>stem</i> .ENF_SIGNAL	ENF signals are supported for this class	CLS.ENF_SIGNAL=1

Table 305. Class descriptor entry data (continued)

Variable	Description	Example
<i>stem</i> .KEY_QUALIFIERS	The number of matching qualifiers RACF uses when loading generic profile names to satisfy an authorization request if a discrete profile does not exist for the resource	CLS.KEY_QUALIFIERS=0
<i>stem</i> .IBM_CLASS	Static IBM class	CLS.IBM_CLASS=1
<i>stem</i> .DYNAMIC	Class is defined using dynamic CDT	CLS.DYNAMIC=0
<i>stem</i> .DUPLICATE	Class is defined in both the static customer CDT (ICHRRCDE) and the dynamic CDT	CLS.DUPLICATE=0
<i>stem</i> .CLASS_ACTIVE	Indicates if SETROPTS CLASSACT is in effect for the class. For the PROGRAM class, indicates if SETROPTS WHEN(PROGRAM) is in effect.	CLS.CLASS_ACTIVE=1
The following variables are set when the <prefix>CLASS_SETTINGS variable has been set to “1” by the caller, and the caller is authorized to extract SETROPTS settings using R_admin.		
<i>stem</i> RACLIST_ACTIVE	Indicates if SETROPTS RACLIST is in effect for the class	CLS.RACLIST_ACTIVE=1
<i>stem</i> STATISTICS_ACTIVE	Indicates if SETROPTS STATISTICS is in effect for the class	CLS.STATISTICS_ACTIVE=0
<i>stem</i> GENERIC_ACTIVE	Indicates if SETROPTS GENERIC is in effect for the class	CLS.GENERIC_ACTIVE=1
<i>stem</i> .GENCMD_ACTIVE	Indicates if SETROPTS GENCMD is in effect for the class	CLS.GENCMD_ACTIVE=1
<i>stem</i> .GENLIST_ACTIVE	Indicates if SETROPTS GENLIST is in effect for the class	CLS.GENLIST_ACTIVE=0
<i>stem</i> GLOBAL_ACTIVE	Indicates if SETROPTS GLOBAL is in effect for the class	CLS.GLOBAL_ACTIVE=0
The following variables are set when the <prefix>CLASS_SETTINGS variable has been set to “1” by the caller, the caller is authorized to extract SETROPTS settings using R_admin, and the caller is authorized to see audit settings.		
<i>stem</i> .LOGOPTIONS	The SETROPTS LOGOPTIONS setting for the class	CLS.LOGOPTIONS=DEFAULT
<i>stem</i> AUDIT_ACTIVE	Indicates if SETROPTS AUDIT is in effect for the class	CLS.AUDIT_ACTIVE=1

Appendix A. ICHEINTY, ICHETEST, and ICHEACTN macros

The ICHEINTY, ICHETEST, and ICHEACTN macros are described in this topic separately from the other interfaces because of their complexity and the cautions required for their use.

Guidelines:

- Use RACROUTE REQUEST=EXTRACT instead of these macros whenever possible. However, only the RACF command processors completely validate the data entering the database. Therefore, it is preferable to use the RACF commands than either ICHEINTY or RACROUTE REQUEST=EXTRACT when updating the database. For more information about RACROUTE REQUEST=EXTRACT, see *z/OS Security Server RACROUTE Macro Reference*.
- In general, always use the RACF commands to create RACF resource profiles. If you use ICHEINTY instead, create profiles that are supported by the command processors. For instance, ICHEINTY allows you to create a fully-qualified generic profile in a general resource class or a data set profile containing characters that are not valid, but those profiles are not supported by the RACF command processors.

You can use the ICHEINTY, ICHETEST, and ICHEACTN macros to locate (retrieve) and update the various profiles on the RACF database.

ICHEINTY

Locates or updates the profile.

ICHETEST

Tests for user-specified conditions on selected fields in the profile.

ICHEACTN

Retrieves or alters specified fields within the retrieved profile.

If you plan on using these macros, you should exercise caution because they:

- Perform only limited parameter validation. The module issuing these macros must be authorized (supervisor state, system key, or APF-authorized).
- Do not pass control to any exit routines except indirectly. If FLDACC=YES was specified on the ICHEINTY macro, the RACROUTE REQUEST=AUTH exits are given control during field access checking.
- Do not do any logging except indirectly. Logging can occur during field access checking if the RACROUTE REQUEST=AUTH request exit requests it.
- Do not complete data consistency checking. For example,
 1. They do not ensure that all fields in a profile have the data expected by subsequent RACF processing.
 2. They do not ensure that related profiles are updated in a consistent manner. For example, a group profile must point to its superior group profile and the superior group must point to the subgroup profile. The command processors would ensure this, but these macros do not.

Note: You should thoroughly familiarize yourself with the template information contained in [Appendix D, “RACF database templates,”](#) on page 607 before you read this topic.

These macros can be used by callers in either 31- or 24-bit addressing mode. The parameter lists can be located above 16MB if the caller is in 31-bit mode.

Application programs must be structured such that a task requesting RACF services does not do so while other I/O initiated by the task is outstanding. If such I/O is required, the task should either wait for the other I/O to complete before requesting RACF services, or the other I/O should be initiated under a separate task. This is necessary to assure proper processing in recovery situations.

ICHEINTY macro

The ICHEINTY macro provides a direct interface to the RACF database through the RACF manager. Its function is to locate or update a profile in the RACF database.

You can use the ICHEINTY macro with the ICHETEST and ICHEACTN macros to test and conditionally update fields in RACF profiles.

The ICHEINTY macro must be issued from a task running in non-cross-memory mode with no locks held. The issuing task must be authorized (APF-authorized, in system key 0-7, or running in supervisor state).

You can reference only one segment with each ICHEINTY call; however, you can access more than one field in a segment using a single call. If you need to retrieve or update more than one segment, issue a separate ICHEINTY for each segment.

When activated, automatic direction of application update propagates ICHEINTY ADD, ALTER, DELETE, DELETEA, and RENAME updates to selected remote nodes. Only ICHEINTY requests with return code 0 are propagated. ICHEINTY requests larger than approximately 4,700 bytes will fail to propagate. Message code IRR116I with reason code 506 will be saved to the RRSFLIST dataset.

If your installation uses RACF remote sharing facility (RRSF) to propagate password changes to other (target) nodes, you should be aware that attempts to change the password of a *revoked* user on a target node fails. Therefore, if your installation implemented automatic password direction or password synchronization, make sure that each target user ID is *resumed* before changing the password.

The format of the ICHEINTY macro definition is:

```
[label] ICHEINTY [operation]
      [,TYPE='GRP' | 'USR' | 'CON' | 'DS' | 'GEN']
      [,ENTRY=entry-address]
      [,ENTRYX=extended-entry address]
      [,CLASS=class-address]
      [,FLDACC=NO | YES]
      [,RELEASE=number | (,CHECK) | (number,CHECK)]
      [,RUN=YES | NO]
      [,ACEE=acee address]
      [,WKSP=subpool number]
      [,CHAIN=parm-list address]
      [,DATAMAP=OLD | NEW]
      [,SEGMENT='segment name']
      [,VOLUME=volume-address]
      [,ACTIONS=(action-address,...)]
      [,TESTS=(test-addr[, [AND], test-addr]...)]
      [,WKAREA=workarea-address]
      [,NEWNAME=newname-address]
      [,NEWNAMX=extended-newname-address]
      [,RBA=rba-address]
      [,FLDEF=fldef-address]
      [,OPTIONS=(list-of-options)]
      [,SMC=YES | NO]
      [,GENERIC=NO | YES | UNCOND]
      [,DATEFMT=YYDDDF | YYYYDDDF]
      [,MF=I | L | (E,address)]
      [,INDEX=ASIS | ONLY | MULTIPLE]
```

operation

Specifies the operation that RACF is to perform on the specified profile. The valid operation values are ADD, ALTER, ALTERI, DELETE, DELETEA, FLDEF, LOCATE, NEXT, NEXTC, and RENAME. This operand is positional and is required if you specify MF=I or MF=L.

Some operations are based on assumptions. If a requested operation violates an assumption, the operation fails.

ADD

Defines entities to RACF by adding profiles to the RACF database. ADD processing does the following:

- Creates a profile for the new entry with fields containing null values. (See [“Using ICHEACTN to alter data when the ICHEINTY has DATAMAP=NEW” on page 513](#) and [“Using ICHEACTN to alter data when ICHEINTY has DATAMAP=OLD” on page 516](#).)

- Alters field values as specified by associated ICHEACTN macro instructions.
- Allocates space for the profile on the RACF database and writes the profile.
- Creates an index entry for the profile. The index entry points to the new profile.

Some considerations regarding ADD are:

- ADD processing assumes that the profile does not already exist. If the profile already exists (the index contains the profile name), any of the following conditions cause a return code of 8 (X'08'):
 - TYPE is not 'DS'
 - TYPE is 'DS' and duplicate data set name creation is prohibited from ICHSECOP.
 - TYPE is 'DS' and one of the existing profiles for the entity name contains in its volume list the volume that is specified by the VOLUME keyword.
- For TYPE='DS', ICHEINTY sets a return code of 60 (X'3C') if VOLUME is not specified and there are multiple profiles for the entity name.
- For TYPE='USR', when automatic direction of application updates and automatic password direction are active, a single ICHEINTY with ACTIONS= that specifies both add user and password information results in the propagation of 2 requests to the target node, one by automatic direction of application updates, the other by automatic password direction. At the target node, these two requests execute using different user IDs and can execute concurrently; this makes the results of the ICHEINTY unpredictable at the target node. The unpredictable results occur for the propagation of any combination of ICHEINTYs (a program with a single ICHEINTY, or multiple ICHEINTYs within the same or different programs) that add a user and specify password information for that user.

Guideline: Use an ADDUSER command or the R-ADMIN function from an application program to define a RACF user under these conditions.

- For TYPE='GEN', you can add the same entity name to any number of different classes. If the class is TAPEVOL, ICHEINTY sets a return code of 60 (X'3C') if a VOLUME is specified but is not an existing TAPEVOL. ICHEINTY creates a profile for a TAPEVOL only when VOLUME is not specified; otherwise, it updates an existing profile. The macro creates an index entry in either case. The result of this special TAPEVOL processing is that RACF maintains only one profile for a multivolume tape. You can refer to that profile by specifying any of the volumes it protects.
- You must supply all the information required by RACF for subsequent processing: for example, owner, creation date.
- In general, use the RACF command processors to create RACF resource profiles. If you use ICHEINTY instead, create profiles that are supported by the command processors. For instance, ICHEINTY allows you to create a fully-qualified generic profile in a general resource class and a data set profile containing characters that are not valid, but those profiles are not supported by the RACF command processors.

ALTER

Alters field values in an existing profile on the RACF database. ALTER processing:

- Locates the profile on the RACF database.
- Performs the tests that the ICHETEST macros specify, if any macros are present. The TESTS operand on the ICHEINTY macro names the tests to be performed.
- Alters field values as specified by associated ICHEACTN macro instructions.
- Writes the profile back to the primary and backup (if active) RACF databases.

Some considerations regarding ALTER are:

- If the profile is too large to be rewritten to the same location in the RACF database, RACF allocates new space, writes the profile to the new location, and updates the index entry for the profile to point to the new location.
- Do not use ICHEINTY to ALTER (or ALTERI) repeat group count fields. These fields are updated automatically whenever a repeat group changes its size. Repeat group count fields can be read.

ALTERI

Similar to the ALTER operation with the following exceptions:

- Fields are updated in place. ICHEINTY sets a return code of 68 (X'44') and fails the operation if the altered profile has a length that differs from the original profile.
- The update to the field occurs with only a shared lock on the RACF database. Therefore, other ALTERI, LOCATE, NEXT, or NEXTC requests can take place simultaneously.
- You can specify the RBA of the level one index block, containing the pointer to the profile to be altered. This improves processing efficiency.
- ALTERI processing writes the profile back to the primary RACF database only; it does not write to the backup, unless you otherwise specified in the data set name table. RACF uses ALTERI to update statistical information in profiles. ALTERI should only be used with fields that are marked in the template as statistical. For more information about the RACF database templates, see Appendix D, “[RACF database templates](#),” on page 607.

DELETE

Deletes a profile from the RACF database. DELETE processing:

- Deletes the index entry for the entity.
- Frees space that is used for the profile on the RACF database.

Some considerations regarding DELETE are:

- You cannot specify the ACTIONS operand with the DELETE operation.
- For TYPE='DS', ICHEINTY sets a return code of 60 (X'3C') if you specified VOLUME and a profile containing the specified volume in its VOLSER list was not found. ICHEINTY sets a return code of 56 (X'38') if you do not specify VOLUME and there are multiple profiles for the entity name.
- ICHEINTY deletes the profile for a TAPEVOL only when the volume being deleted is the last in its set. Otherwise, the macro deletes the index entry and removes the volume from the VOLSER list.

DELETEA

Deletes all the members of a TAPEVOL set from the RACF database. It is similar to the DELETE operation with the following exception:

- If you specify 'TYPE=GEN' and the class is TAPEVOL, ICHEINTY deletes the profile along with the index entry for each volume in the set.

FLDEF

Builds the area that the FLDEF operand uses. The area contains control information and the list that is generated by the TESTS and ACTIONS operands.

Some assumptions and considerations regarding FLDEF are:

- FLDEF creates a separate area for ICHEACTN and ICHETEST pointers, which can be referenced from one or more ICHEINTY macros.
- You can maintain the field definition area with the MF=E form of ICHEINTY FLDEF.
- When referencing the field definition area from a remote ICHEINTY, specify FLDEF=field-definition-area, and do *not* specify any of the ACTION, FLDEF, TESTC, and TESTM options on the OPTIONS keyword.

LOCATE

Retrieves zero or more fields from an existing RACF profile in the RACF database. LOCATE processing:

- Locates the profile in the RACF database.
- Performs the tests that the ICHETEST macros specify, if any macros are present. The TESTS operand on the ICHEINTY macro names the tests to be performed.
- Retrieves field values as specified by associated ICHEACTN macro instructions into the caller-specified work area.

Some assumptions and considerations regarding LOCATE are:

- ICHEINTY sets a return code of 44 (X'2C') if the values being returned are too large for the work area that is provided and you did not specify the WKSP operand, which would have provided you with an additional work area.
- For TYPE='DS', ICHEINTY set a return code of 60 (X'3C') if you specify VOLUME and one or more profiles were found for the data set name but none contained the specified volume name in its VOLSER list. ICHEINTY sets a return code of 56 (X'38') if you do not specify VOLUME and there are multiple profiles for the entity name.
- ICHEINTY sets a return code of 52 (X'34') if an ICHETEST macro specified by the TESTS operand failed. The LOCATE operation terminates at this point.

NEXT

Retrieves zero or more fields from the profile whose name follows the name specified by the ENTRY or ENTRYX operand. The NEXT operation updates the area pointed to by the ENTRY or ENTRYX operand with the name of the profile just completed. NEXT processing:

- Locates the profile of the first entity of the specified type that follows the specified entity in the RACF database.
- Performs the tests that the ICHETEST macros specify, if any macros are present. The TESTS operand on the ICHEINTY macro names the tests to be performed.
- Retrieves field values as specified by associated ICHEACTN macro instructions into the caller specified work area.

Some considerations regarding NEXT are:

- If the entity retrieved has the same name as the entity that follows it in the RACF database, ICHEINTY sets the duplicate data set name count. The count becomes 2 if it was zero on entry; otherwise, the count increases by one. The count is zero if the entity is not a duplicate of the one that follows it.
- ICHEINTY sets a return code of 44 (X'2C') if the values being returned do not fit into the provided work area, unless you specified WKSP which would provide you with an additional work area.
- For qualified types (data set, general, and connect), the located entity must have the same high-level qualifier as the specified entity. Otherwise, the macro sets a return code of 12 (X'0C').
- For data set profiles, the qualifier includes the first period in the name.
- For TYPE='DS', if the duplicate data set name count in the work area is not zero, ICHEINTY locates the specified data set name. (That is, if the duplicate data set count equals N, then the macro locates the Nth occurrence of the specified name. If there are less than N occurrences of the specified name, the same process occurs as when the duplicate data set count is zero; ICHEINTY locates the profile of the first entity of the specified type that follows the specified entity in the RACF data set.) If you want to locate the first DATASET profile with a name greater than or equal to the name specified by ENTRY= or ENTRYX, you can do so by setting the duplicate data set name count to one.
- If an ICHETEST macro specified in the TESTS operand failed, ICHEINTY sets a return code of 52 (X'34') and the NEXT operation terminates.
- If you specify a segment other than BASE on this ICHEINTY, or on any ICHEINTY in a chain, the RACF manager skips any profiles that do not contain an occurrence of the segment. Normal processing (TESTS on ICHEINTY, ICHEACTN) resumes with the next profile containing the segment. This simplifies the process of finding users who are defined to TSO, for example.

NEXTC

is similar to the NEXT operation with the following exception:

- For qualified types (data set, general, and connect), ICHEINTY does not make the high-level qualifier check. For unqualified types (group and user), NEXTC processing is identical to NEXT processing. The qualifier for a general profile is the 8-character class name.

Guideline: Do not use NEXTC with general resource classes, because it might have unpredictable results. When you reach the end of the profiles for the class that you specified with the CLASS

keyword, ICHEINTY retrieves the first profile for the next general resource class that has profiles. However, you have no way of determining what class that profile belongs to, and in some cases you might not even know that ICHEINTY switched to a new class.

RENAME

renames a data set, SFS file, or SFS directory entry in the RACF database.

You must specify the NEWNAME or NEWNAMX operand.

When renaming resources in the FILE and DIRECTRY class, use ENTRYX and NEWNAMX.

TYPE= 'GRP' | 'USR' | 'CON' | 'DS' | 'GEN'

Specifies the type of the entry as GROUP ('GRP'), USER ('USR'), CONNECT ('CON'), DATASET ('DS'), or general resource ('GEN').

The final parameter list the SVC uses as a request to the RACF manager must include a value for TYPE.

ENTRY= entry-address

Specifies the address of a 1-byte entry name length field followed by the entry name. The NEXT and NEXTC operations update this field. When using NEXT or NEXTC you should initialize the field in this way. To initialize the entry field, point to a field that has a length of 1 and a field of X'00'. The area that is pointed to must allow for 255 bytes of data to be returned.

The final parameter list the SVC uses as a request to the RACF manager must include a value for ENTRY or ENTRYX.

ENTRYX= extended-entry-address

Specifies the extended entry name field for long name support. You must specify the address of *two* 2-byte fields, followed by the entry name.

- The first 2-byte field specifies a buffer length which can be from 0 to 255 bytes. This length field only refers to the length of the buffer that contains the entity name; it does not include the length of either length field.
- The second 2-byte field specifies the actual length of the entity name. This length field includes only the length of the actual name without any trailing blanks; it does not include the length of either length field.

As with ENTRY, the NEXT and NEXTC operations update this field. The area that is pointed to must allow for a name which is of maximum entry length. ENTRY and ENTRYX are mutually exclusive keywords.

Guideline: Use the ENTRYX keyword to save storage, because it allows you to code to the specific amount of space that you need.

The final parameter list the SVC uses as a request to the RACF manager must include a value for ENTRY or ENTRYX. To use the ENTRYX keyword, you must specify RELEASE=1.9.

CLASS= class-address

Specifies the address of an 8-character class name (left-justified and blank-padded, if necessary.) The class name is required when TYPE='GEN' and is ignored for all other types.

FLDACC= NO | YES

Specifies the presence or absence of field level access checking. If you specify FLDACC=YES, the RACF database manager checks to see that the user running your program has the authority to retrieve or modify the fields that have been specified in the ICHETEST and ICHEACTN macros associated with the current ICHEINTY macro.

Notes:

1. For field level access checking to occur, you must specify RELEASE=1.8 or later when you code the ICHEINTY and associated ICHETEST and ICHEACTN macros. RACF bypasses field access checking for any ICHETEST or ICHEACTN macro for which RELEASE=1.8 or later has not been specified. In addition, before your program executes, the security administrator must activate the FIELD class and process the FIELD class using SETROPTS RACLIST. If you code FLDACC=YES and the field class is not active and has not been processed using SETROPTS RACLIST, the request fails with a return code of 60.
2. In addition, the security administrator must issue the RDEFINE and PERMIT commands to designate those users who have the authority to access the fields designated in the ICHETEST and ICHEACTN macros.
3. If you specify FLDACC=NO or omit the parameter, the manager does not perform field level access checking.
4. If you specify FLDACC=YES for the ADD, ALTER, or LOCATE operation with segment name CSDATA and field name CSCDATA or CSKEY, the data associated with the CSKEY field (if present in the ICHEINTY parameter list) is used instead of the template name for field level access checking. This allows field level access checking to use the FIELD profile USER.CSDATA.*custom-field-name* or GROUP.CSDATA.*custom-field-name* for authorization, similar to the authorization that occurs for the RACF commands.

RELEASE=number

RELEASE=(,CHECK)

RELEASE=(number,CHECK)

Specifies the release number. The release numbers that you can specify with the ICHEINTY macro are PLV0001, 77B0, 77A0, 7790, 7780, 7770, 7760, 7750, 7740, 7730, 7720, 7709, 7708, 7707, 7706, 7705, 7703, 2608, 2.6, 2.4, 2.3, 2.2, 2.1, 1.9.2, 1.9, 1.8.1, 1.8, or 1.7.

When you specify 1.8 or later, the RACF manager returns data using the 1.8 user work area format (documented in the sections [“Using ICHEACTN to retrieve data when ICHEINTY has DATAMAP=NEW”](#) on page 511 and [“Using ICHEACTN to retrieve data when the ICHEINTY has DATAMAP=OLD”](#) on page 514). In effect, DATAMAP defaults to DATAMAP=NEW if you specify RELEASE=1.8 or later and omit DATAMAP.

If you specify RELEASE=1.7, or allow the release parameter to default to 1.7, the RACF manager returns data using the 1.7 user work area format. In this case, DATAMAP defaults to DATAMAP=OLD if you omit it.

If you want to use 1.8 parameters, and the 1.7 user work area format, you must specify RELEASE=1.8 or later and DATAMAP=OLD.

To use the 1.8 parameters, you must specify RELEASE=1.8 or later. If you specify RELEASE=1.8 or later the ICHEINTY parameter list must be in modifiable storage.

The default is RELEASE=1.7.

Table 306. ICHEINTY parameters		
Parameter	RELEASE=1.7 and earlier	RELEASE=1.8 or later
ACEE		X
ACTIONS	X	X
CHAIN		X
CLASS	X	X
DATAMAP		X
ENTRY	X	X
FLDACC		X

<i>Table 306. ICHEINTY parameters (continued)</i>		
Parameter	RELEASE=1.7 and earlier	RELEASE=1.8 or later
FLDEF	X	X
GENERIC	X	X
INDEX		X
MF	X	X
NEWNAME	X	X
OPTIONS	X	X
RBA	X	X
RELEASE	X	X
RUN		X
SEGMENT		X
SMC	X	X
TESTS	X	X
TYPE	X	X
VOLUME	X	X
WKAREA	X	X
WKSP		X

RUN= YES | NO

Specifies whether to activate or deactivate the parameter list. If you specify RUN=NO, the RACF manager ignores the request designated by this ICHEINTY macro although it processes the CHAIN parameter. If you specify RUN=YES, RACF processes this request and also processes the CHAIN parameter. Thus you can use the RUN parameter to deactivate or activate one or more ICHEINTY parameter lists without having to rearrange the chaining.

ACEE= acee-address

Specifies an ACEE that RACF uses to perform field authorization checking. If you specify FLDACC=YES, but omit ACEE, the RACF manager uses the appropriate ACEE pointed to either by the TCB or the ASXB.

WKSP= subpool-number

Specifies the number of a storage subpool. If the area specified by the WKAREA parameter is too small to contain the field data that is retrieved from LOCATE, NEXT, or NEXTC operation, the RACF manager obtains an additional work area from this subpool in the callers key. The RACF manager returns the address of the work area in the fullword at offset 60 (X'3C') in the ICHEINTY parameter list. If additional storage is not needed, the RACF manager sets the fullword to zero. It is your responsibility to free any returned work area; its subpool number is stored in the one-byte field at offset 29 (X'1D') in the parameter list, its address in the fullword at offset 60 (X'3C') in the ICHEINTY parameter list and its size in the first fullword of the work area itself.

Notes:

1. Even if you specify WKSP, you must still provide a work area at least 30 bytes long using the WKAREA operand.
2. If the RACF manager is unable to obtain a large enough work area, an "out-of-storage" abend occurs.

3. WKSP does not apply for ICHEINTY requests with INDEX=MULTIPLE.

Guidelines:

- Simplify your coding by specifying WKSP and you can avoid processing a return code of 44 (X'2C').
- Carefully select a subpool because MVS processes certain subpools differently. In particular, your results might be unpredictable if you specify subpool 0, subpool 250, or any subpool, such as 227–231 or 241, that is documented in *z/OS MVS Programming: Assembler Services Guide* as having a storage key of USER.

CHAIN= parm-list address

Specifies a parameter list that another ICHEINTY macro created. This chained parameter list executes after the current one within the same manager invocation. If several ICHEINTY requests pertain to the same profile, you can use CHAIN to string the requests together. This chaining improves performance because the RACF manager retrieves the profile only once for the entire chain. Each ICHEINTY parameter list in the chain must contain the same values for the following parameters: ACEE, CLASS, ENTRY or ENTRYX, GENERIC, RBA, SMC, TYPE, and VOLUME.

Notes:

1. Because there are no connect profiles in the database, chained ICHEINTY requests do not work as anticipated. Therefore, if you chain two ICHEINTY requests with TYPE=CON, the second ICHEINTY is ignored.
2. You cannot specify NEWNAME for any request in the chain.
3. When you chain ICHEINTY macros together, they must obey the following rules:
 - The first ICHEINTY in the chain must be a LOCATE, NEXT/NEXTC, ALTER/ALTERI, ADD, or a DELETE with SEGMENT specified.
 - The remaining ICHEINTY macros in the chain must be:
 - LOCATE, if the first was LOCATE
 - NEXT/NEXTC, if the first was NEXT or NEXTC
 - ALTERI, if the first was ALTERI
 - ALTER, if the first was ALTER, DELETE, or ADD
 - DELETE, with SEGMENT, if the first was ALTER or DELETE
 - The RACF manager return code is set to the highest return code of any of the individual ICHEINTY macros that have been chained together.
 - For chained ICHEINTY parameter lists, within the bounds of any one chain of ICHEINTYs, an alias field (indicated by the alias bit in the fields template definition) can be referenced, either directly or indirectly (for example, delete of a segment containing a defined alias field, or delete of a profile with a segment containing a defined alias field) multiple times. If there are multiple references, there can be tests on any of these references except for the last reference. If a parameter list chain does not meet this requirement, return code 36 (X'24') reason code 11 (X'B') is returned.

If an FLDEF test is specified for a delete or alter ICHEINTY whose last action on a specific alias field lacks a test, then unless there is an ICHEINTY parameter list earlier in the chain that also references the same alias field, the FLDEF test does not count as a test on the last alias reference. This is because if the FLDEF test on the ICHEINTY fails, none of the actions in the ICHEINTY parameter list is executed.

DATAMAP= OLD | NEW (default depends on RELEASE)

DATAMAP determines the format of the work area returned for a LOCATE, NEXT, or NEXTC operation and the format of the data you provide when you issue an ICHEINTY request to update the database. You can specify the DATAMAP parameter in several different combinations to tailor it to your system:

- If you specify RELEASE=1.7 or allow RELEASE= to default to 1.7, you need not specify DATAMAP. It defaults to OLD, meaning the 1.7 format.

Note: You cannot specify DATAMAP=NEW if you specify RELEASE=1.7 or default to it.

- If you specify RELEASE=1.8 or later, and allow DATAMAP to default, it defaults to NEW, meaning the 1.8 format.
- If you specify RELEASE=1.8 or later, and want to use the 1.7 user work area format, you must specify DATAMAP=OLD for the data to be retrieved in the 1.7 format.

Note: Releases before 1.7 are in the 1.7 format.

SEGMENT= 'segment name'

Specifies that this request is to apply to a specific segment in the profile. If you do not specify a specific segment, the default is the BASE segment. If you specify a segment other than the BASE segment, the operation cannot be ADD, DELETEA, or RENAME. If you specify a segment, the DELETE operation deletes only that segment. If you do not specify a segment, the DELETE operation deletes the entire profile. If you specify a segment, the ALTER operation alters only that segment. If you do not specify a segment, the ALTER operation updates the BASE segment. See [Appendix D, "RACF database templates," on page 607](#) for a list of valid segment names.

VOLUME= volume-address

Specifies the address of a 6-character volume identifier. When TYPE='DS', the volume identifier differentiates among data sets with the same name. When the operation is ADD, TYPE='GEN', and the class is TAPEVOL, the volume identifier specifies the name of an existing tape volume set to which the current entry is to be added. In all other cases, ICHEINTY ignores the volume identifier.

ACTIONS= (action-address,.....)

Specifies the address of one or more ICHEACTN macros that determine which profile fields the RACF manager is to retrieve or update. For a description of the ICHEACTN macro, see ["ICHEACTN macro" on page 508](#). You can specify up to 255 actions.

Note: If you specify ACTION on the execute form of the ICHEINTY macro, the number of actions you specify should agree with the number of actions you have specified on the list form of the macro (or the FLDEF list, if you use that instead). If the numbers do not match, you must specify the OPTION keyword on the execute form to update the counts, using the appropriate ACTION operands.

TESTS= (test-addr[, [AND], test-addr]...)

Allows some preliminary testing on selected conditions before the execution of the operation specified by the ICHEINTY macro. You must specify an odd number of items (including the connector 'AND'). Each address must be the address of a list that is built by the ICHETEST macro. See ["ICHETEST macro" on page 505](#).

Note: If you specify TEST on the execute form of the ICHEINTY macro, the number of tests you specify should agree with the number of tests you have specified on the list form of the macro (or the FLDEF list, if you use that instead). If the numbers do not match, you must specify the OPTION keyword on the execute form to update the counts, using the appropriate TESTC or TESTM operands.

WKAREA= wkarea-address

Specifies the address of the area into which the retrieved values are to be placed. This operand is valid and required only for the LOCATE, NEXT, and NEXTC operations. The work area must be at least 30 bytes long.

When you specify INDEX=MULTIPLE, the work area must be at least 4K long. See [Table 308 on page 499](#) for the format of the work area for INDEX=MULTIPLE.

For a related operand, see the description of WKSP.

Table 307. Format for the ICHEINTY work area when INDEX=MULTIPLE is not specified

Offset (hex)	Length	Description
0	4	Length of entire work area
4	6	RBA return area
A	1	Flags
B	1	Reserved
C	4	Duplicate data set name count
10	8	Reserved
18	4	Length of data returned into work area
1C	variable	Field value return area

Table 308. Format for the ICHEINTY work area when INDEX=MULTIPLE is specified

Offset (hex)	Length	Description
0	4	Length of entire work area
4	20	Reserved
18	4	Length of data returned into work area
1C	2	Number of returned profile names
1E	1	Flag: Bit Meaning 1 No more profiles after this set.
1F	1	Reserved
20	variable	Start of list of returned profile names. Format of returned profile name: Length Description 1 Length of profile name variable Profile name

NEWNAME= newname-address

Specifies the address of the new name to be assigned to the entity named by the ENTRY operand. The name must be left-justified and followed by at least one blank.

Whereas ENTRY is a 1-byte length field followed by a name, NEWNAME specifies an entry name that is *not* preceded by a 1-byte length field. This operand is valid only for the RENAME operation.

When renaming resources in the FILE and DIRECTORY class, use ENTRYX and NEWNAMX. If NEWNAME is used, the name must be 39 characters long or less.

NEWNAMX= extended-newname-address

Specifies the address of the new name to be assigned to the entity in the ENTRYX keyword. The format of the new name is the same as that of the ENTRYX keyword.

- The first 2-byte field specifies a buffer length that can be from 0 to 255 bytes. This length field refers only to the length of the buffer that contains the entity name; it does not include the length of either length field.
- The second 2-byte field specifies the actual length of the entity name. This length field includes only the length of the actual name without any trailing blanks; it does not include the length of either length field.

NEWNAMX and NEWNAME are mutually exclusive parameters, as are NEWNAMX and WKAREA. The NEWNAMX keyword is valid only for the RENAME operation. To use NEWNAMX, you must specify RELEASE=1.9 or later.

RBA= RBA-address

Specifies the address of a 6-byte relative byte area (RBA) of a level one index that points to the profile to be altered. This keyword is valid only for an ALTERI request. RBA should specify the value returned by a previous LOCATE operation.

FLDEF= fldef-address

Specifies a remote list of ACTION/TEST pointers set up by an ICHEINTY with the FLDEF operation.

OPTIONS= (list-of-options)

Provides more direct control of the code generated by the EXECUTE form of the macro. This operand is valid only with the EXECUTE form of the macro.

Specify one or more of the following options, separated by a comma:

NOPRO

Does not generate any prologue code; that is, the instructions that set the type of request, such as ADD, by updating the first two bytes of the parameter list, are not generated.

FLDEF

Generates the FLDEF pointer relocation code to point to the list of ACTION and TEST pointers in the ICHEINTY macro expansion.

ACTION

Generates code to set the number of ACTIONs that are to be performed.

TESTC

Generates code to set the number of TESTs that are to be performed.

TESTM

Generates code to set both actual and maximum number of TESTs.

NOEXEC

Does not generate the SVC instruction to invoke the RACF manager. This subfield is useful with the EXECUTE form of the macro to allow partial setup of the parameter list.

SMC= YES | NO

Controls the 'set-must-complete' operation mode of the RACF manager. YES is the default mode of operation.

Note: If an ICHEINTY request is propagated by automatic direction of application updates, the SMC=NO keyword is not propagated. The ICHEINTY request always runs as SMC=YES on the target node.

GENERIC= NO | YES | UNCOND

Informs the RACF manager whether the given entity name is a generic name.

NO

Never generic.

The RACF manager does not attempt to convert the name specified by the ENTRY operand from external to internal form. GENERIC=NO is the default.

YES

Can be generic.

The RACF manager attempts to convert the name specified by the ENTRY operand from external to internal form. The RACF manager does the conversion only if the entity name contains a generic character (an * or %). If the entity name does not contain a generic character, processing continues without any conversion.

UNCOND

Always generic.

The RACF manager unconditionally converts the name specified by the ENTRY operand from external to internal form.

For RENAME, the same process applies also to the NEWNAME operand.

DATEFMT= YYYYDDDF | YYDDDF

Specifies the format of the date that you want to extract or replace.

If you specify DATEFMT=YYYYDDDF with a LOCATE, NEXT, or NEXTC operation, RACF retrieves date fields in the format *ccyydddF*, where *cc*=19 or *cc*=20. If an ADD, ALTER, or ALTERI operation is specified, RACF accepts dates in the format *ccyydddF*, where *cc*=19 or *cc*=20, unless the data being retrieved is in an uninitialized state in the RACF database, in which case 0000000F or FFFFFFFF is returned. When accepting a date as input to place into the database, RACF validates that *cc* is 19 or 20 and that:

- For *cc*=19, 70 < *yy* <= 99
- For *cc*=20, 00 <= *yy* <= 70

If you specify DATEFMT=YYDDDF, RACF retrieves and accepts dates in the three-byte format.

To specify the DATEFMT keyword, you must specify Release=1.9.2 or a later release number.

DATEFMT=YYDDDF is the default.

MF= I | L | (E,address)

Specifies the form of the macro as either INLINE, LIST, or EXECUTE.

The INLINE form generates code to branch around the parameter list. In the MF=I form, the label names the instruction preceding the parameter list. MF=I is the default.

The LIST form reserves and initializes storage.

The EXECUTE form modifies a list that is defined elsewhere. If you use the EXECUTE form, you must specify the address of the list to be modified. The address can be an A-type address or register (2 through 12).

INDEX= ASIS | ONLY | MULTIPLE

Specifies how RACF processes the generic profile names in the index blocks of the RACF database.

The INDEX option is supported only for NEXT requests. Do not specify INDEX with GENERIC=NO or with TYPE values other than GEN or DS.

ASIS

Specifies normal RACF processing. This is the default value.

ONLY

Specifies that RACF should return the name of the next generic profile with the same HLQ, or the next generic profile for the same class, but should not return the content of any profile.

MULTIPLE

Specifies that RACF should return more than one generic profile name. RACF returns the name of the next generic profile with the same HLQ or the next generic profile for the same class, and the names of subsequent generic profiles in the same level 1 index block with the same HLQ or class name.

With INDEX=MULTIPLE, you must specify WKAREA to supply a work area that is at least 4K in length to hold the returned profile names. See [Table 308 on page 499](#) for the work area format.

Note: RACF places the last returned name in the ENTRY or ENTRYX field, as it normally does for NEXT operations.

Return codes from the ICHEINTY macro

If you did not specify RELEASE=1.8 or later, Register 15 contains the ICHEINTY return code and Register 0 contains the reason code. If you specified RELEASE=1.8 or later, Register 15 contains the highest return code from any of the ICHEINTY macros; Register 0 contains the corresponding reason code. The return code for each ICHEINTY macro appears in the fullword at offset 52(X'34') in each ICHEINTY parameter list; the corresponding reason code appears in the fullword at offset 56(X'38').

Hex (Dec)

Description

0 (0)

The requested operation was successful.

4 (4)

If the reason code is 0, a recovery environment could not be established; if the reason code is 4, an invalid function code was specified. (Valid functions are RACLIST, RACXTRT, and ICHEINTY. The parameter list was not valid for any of those functions.)

8 (8)

An attempt was made to add an entry to the RACF database but an identical entry already exists.

C (12)

For requests other than NEXT or NEXTC, the specified entry did not exist.

For NEXT or NEXTC requests, no subsequent entries satisfied the request.

10 (16)

Reserved for IBM's use.

14 (20)

The RACF database did not contain enough space to satisfy the request.

18 (24)

An I/O error occurred while accessing the RACF database.

1C (28)

RACF was not active at the time of the request.

20 (32)

The request type requires a user work area but the area was not provided (the address in the parameter list was 0) or for a RENAME, NEWNAME, or NEWNAMX were not supplied.

24 (36)

The input parameter list or the associated ACTION and TEST blocks contain an error. When this code is returned, the possible reason codes are:

Hex (Dec)

Explanation

1 (1)

The entry name or new name is not valid.

- 2 (2)**
Action specified with DELETE or DELETED.
- 3 (3)**
An action or test specified for an undefined field.
- 4 (4)**
Test specified with RENAME.
- 5 (5)**
Reserved for IBM's use.
- 6 (6)**
Reserved for IBM's use.
- 7 (7)**
Incorrect entry type.
- 8 (8)**
GROUP=YES specified for an ICHEACTN, but the data length given was too long for the associated data. This reason code can occur with DATAMAP=OLD.
- 9 (9)**
GROUP=YES specified for an ICHEACTN, but the data length given was too short for the associated data.
- A (10)**
Chained ICHEINTY macros have inconsistent parameters: (CLASS, ENTRY, ENTRYX, GENERIC, RBA, SMC, TYPE, or VOLUME).
- B (11)**
Chained ICHEINTY macros have inconsistent request types (operations).
- C (12)**
All ICHEINTY macros specified RUN=NO.
- D (13)**
Operation not allowed with SEGMENT keyword.
- E (14)**
Incorrect field specified for GROUP=YES, must be a repeat group count field.
- F (15)**
More than 1000 ICHEINTY macros present in the chain.
- 10 (16)**
Specified SEGMENT name not allowed for the specified profile type.
- 11 (17)**
GROUP=YES specified for an ICHEACTN but the data length given was too long for the associated data. This reason code can occur with DATAMAP=NEW.
- 12 (18)**
Data byte specified on ICHEACTN exceeded the length of the specified fixed-length field.
- 13 (19)**
Inconsistency between action data length and repeat group fields. GROUP=YES data is too short.
- 14 (20)**
Invalid ENTRYX. Current length is greater than 44 and either the primary or the backup database is not in the restructured database format.
- 15 (21)**
Invalid NEWNAMX. Current length is greater than 44 and either the primary or the backup database is not in the restructured database format.
- 16 (22)**
Data length specified on the ICHEACTN macro was less than zero and FLDATA=DEL or FLDATA=COUNT were not specified.
- 17 (23)**
Reserved for IBM's use.

18 (24)

Reserved for IBM's use.

19 (25)

Number of tests is greater than 254.

1A (26)

Invalid date supplied on an ICHEACTN when DATEFMT=YYYYDDDF is specified. Date must have a length of 4 bytes and be in the form CCYYDDDF where CC=19, 70 < YY <= 99 and CC=20, 00 <= YY <= 70.

1B (27)

Repeat count cannot be updated when GROUP=NO is specified.

1C (28)

Alias locate requested but database is stage 0 or 1.

1D (29)

Invalid alias locate IPL.

1E (30)

Alias locate requested for a non-alias field.

1F (31)

Base pointer for test is 0 on an alias locate request.

20 (32)

Alias name length is 0 or greater than 252 on an add, alter, or delete request.

21 (33)

INDEX=MULTIPLE was specified but the work area (WKAREA) is not 4K or longer.

28 (40)

The maximum profile size (65 535 bytes) has been reached; the profile cannot be expanded.

2C (44)

The user-supplied work area was not large enough to hold all the data returned. The work area is filled with data up to, but not including, the first field that did not fit. If WKSP was specified, the manager obtains a new work area, retrieves the data, and sets the return code to 0.

30 (48)

The user-supplied work area was smaller than the minimum amount required (30 bytes).

34 (52)

A test condition specified in the TESTS keyword of the ICHEINTY macro was not met; further processing was suppressed.

38 (56)

You requested an operation on a DATASET type entry that has multiple RACF definitions, but you did not specify a VOLUME to single out a specific entry.

3C (60)

For DATASET type entries, you specified a VOLUME that did not exist in the volume list of any entry with the specified name. For TAPEVOL class entries, a request tried to add a new TAPEVOL to a nonexistent tape volume set.

40 (64)

You attempted to delete one of the entries supplied by IBM (such as SYS1 or IBMUSER) from the RACF database.

44 (68)

An ALTERI request attempted to change the size of the profile being updated.

48 (72)

A request to add an entry to the RACF database would have caused the RACF index to increase to a depth that RACF does not support. The maximum depth is 10 levels.

4C (76)

ICHEINTY encountered an invalid index block or read a non-index block when it expected an index block.

50 (80)

An attempt was made to update one of the following (by a request other than ALTERI):

- The RACF database that has been locked by a RACF utility.
- The RACF database from a system that is in read-only mode (in a RACF sysplex data sharing environment).

54 (84)

Reserved for IBM's use.

58 (88)

At least one (but not all) ICHEACTN macros for information retrieval failed to be executed because of a profile field access violation.

5C (92)

All ICHEACTN macros for information retrieval failed to be executed because of a profile field access violation.

60 (96)

An ICHEACTN macro attempted to alter a field and failed because of a profile field access violation. All ICHEACTN macros for the ICHEINTY were suppressed. For FLDACC entries, the field class may not be active and processed by SETROPTS RACLIST.

64 (100)

The RELEASE keyword on the eForm ICHEINTY specified a release of 1.8 or later and CHECK, but the L-form did not specify a release of 1.8 or later.

68 (104)

The requested profile on the database contains erroneous data. A reason code is returned as follows:

1

The profile is physically too short to contain the data implied by variable field lengths or repeat group count fields.

6C (108)

The RACF manager has been invoked recursively, and an exclusive reserve/enqueue is required. However, a shared reserve/enqueue is already held.

70 (112)

The RACF manager received an unexpected return code from a reserve/enqueue. The reserve/enqueue return code is passed back in register 0.

74 (116)

The maximum length of extended entry of ICHEINTY parameter list is not enough to contain a found profile name.

78 (120)

Reserved (used internal to RACF).

7C (124)

Reserved (used internal to RACF).

80 (128)

This is a data sharing mode return code. A coupling facility function had a problem with the ICB.

84 (132)

A request to expand an alias index entry beyond its maximum size has been denied.

88 (136)

Internal error during encryption of a field.

ICHETEST macro

The ICHETEST macro tests for user-specified conditions on selected data in a RACF profile. You can use the ICHETEST macro with the ICHEINTY and ICHEACTN macros to ensure that a specific requirement is met before processing of the ICHEINTY or ICHEACTN macro occurs. Failure to meet the requirements

that are specified on the ICHETEST macro causes further processing of the associated ICHEINTY or ICHEACTN macro to be suppressed.

The ICHETEST macro must be issued from a task running in non-cross-memory mode with no locks held. The issuing task must be authorized (APF-authorized, in system key 0-7, or running in supervisor state).

The format of the ICHETEST macro is:

```
[label] ICHETEST FIELD=field-name | address
        ,FLDATA=(length,address)
        [,COND=EQ | NE | GT | LT | GE | LE | ONES | ZEROS | MIXED]
        [,ENCRYPT=TEMPLATE | YES | NO]
        [,MF=L | (E,address) | I]
        [,RELEASE=number | (,CHECK) | (number,CHECK)]
```

FIELD= field-name | address

Specifies the field-name in the RACF profile whose value is to be tested.

If you use the LIST form of the macro, specify the name of the field. The name must be from 1 to 8 characters long, not enclosed in quotation marks, and defined in the RACF template. In addition, the field cannot be a combination field name (such as ACL in the group profile). Note, however, that a combination field that specifies only one associated field is allowable. Such a combination field is called an alias field such as OWNER in the GROUP profile.

If you use the EXECUTE or INLINE form of the macro, specify the address of the field name to be tested. The address can be an A-type address or register (2 through 12). For EXECUTE and INLINE, you can also specify the field name as a constant (for example, 'OWNER').

FLDATA= (length,address)

Specifies the data to be tested against.

The length must be greater than zero and less than or equal to the length of field-name in the FIELD operand, or the test fails. For fixed length fields, you can specify a length that is less than the actual length of the field in the profile. For flag fields, the length specified is ignored and a 1-byte length is assumed. For variable-length fields, if the length is not equal to the field length in the profile, the test fails unless COND=NE is specified. Also, for variable-length fields the field data must not contain a length byte.

COND= EQ | NE | GT | LT | GE | LE | ONES | ZEROS | MIXED

Specifies the relationship that must exist between the FLDATA and FIELD values to satisfy the test. For example, COND=GE specifies that the value of FLDATA must be equal to or greater than the value of FIELD.

EQ, NE, GT, LT, GE, and LE are valid only for fixed length or variable-length fields. They are not valid for flag fields.

ONES, ZEROS, and MIXED are valid only for flag fields.

If you omit this operand, COND=EQ is the default. An explanation of ONES, ZEROS, and MIXED follows:

ONES

If the 1 bits exist in the FIELD value base where the 1 bits exist in the FLDATA value, the test is successful.

ZEROS

If the 0 bits exist in the FIELD value where the 1 bits exist in the FLDATA value, the test is successful.

MIXED

If both 0 bits and 1 bits exist in the FIELD value where 1 bits exist in the FLDATA value, the test is successful.

You can think of this operation as being equivalent to doing a Test-Under-Mask operation. The ICHETEST data would be used as the mask, and the profile field would be used as the data.

ENCRYPT= TEMPLATE | YES | NO

Specifies whether the data specified by FLDATA is to be encoded before the test is performed. If ENCRYPT=YES, the data is encoded regardless of whether the template flag associated with the field specifies that it is to be encoded. If ENCRYPT=NO, RACF does not encode the data regardless of the template flag value. If ENCRYPT=TEMPLATE, the template flag determines whether the data is encoded.

ENCRYPT is ignored if you specify COND as ONES, ZEROS, or MIXED.

MF= L | (E,address) | I

Specifies the form of the macro as either LIST, EXECUTE, or INLINE.

The LIST form reserves and initializes storage. MF=L is the default.

The EXECUTE form modifies a list defined elsewhere. If you use the EXECUTE form, you must specify the address of the list to be modified. The address can be an A-type address or register (2 through 12).

The INLINE form is similar to a STANDARD form, except that it generates code to branch around the parameter list. In the MF=I form, the label names the first location of the parameter list, not the preceding instruction.

RELEASE=number**RELEASE=(,CHECK)****RELEASE=(number,CHECK)**

Specifies the release number. The release numbers you can specify with the ICHETEST macro are PLV0001, 77B0, 77A0, 7790, 7780, 7770, 7760, 7750, 7740, 7730, 7720, 7709, 7708, 7707, 7706, 7705, 7703, 2608, 2.6, 2.4, 2.3, 2.2, 2.1, 1.9.2, 1.9, 1.8.1, 1.8, or 1.7.

The default is RELEASE=1.7.

<i>Table 309. ICHETEST parameters</i>		
Parameter	RELEASE=1.7 and earlier	RELEASE=1.8 or later
COND=	X	X
ENCRYPT=	X	X
FIELD	X	X
FLDATA	X	X
MF	X	X

Some considerations regarding the ICHETEST macros are:

- You cannot use the ICHETEST macro with an ICHEINTY macro that has the RENAME operation specified.
- A profile can contain repeat groups. A repeat group consists of one or more sequential fields that can be repeated in the profile. By specifying COND=EQ, you can select the occurrence of the repeat group to which the action applies.

By specifying COND=NE, you can position yourself past the last occurrence of the repeat group. Then you can add a new occurrence to the end of that repeat group with an ICHEACTN macro.

Note: When the ICHEACTN macro refers to a repeat group and more than one ICHETEST macro is specified, the last ICHETEST macro serves to position data retrieval from the profile. Therefore, the last ICHETEST should refer to the same repeat group as the last ICHEACTN; otherwise, the retrieved data is from the last tested field. On multiple tests with fields in repeat groups, each test is processed separately, and if all succeed, the tests are considered to have succeeded.

- Tests involving negative numbers cause unpredictable results.
- If a specified address equals zero, ICHETEST makes no test.

- Use only COND=EQ or COND=NE to test encrypted fields. Other comparisons cause unpredictable results.
- The expansion of the ICHETEST macro MF=L or MF=I includes at offset 1, a 1-byte field whose value is X'00' if the test was successful, or X'01' if the test failed. The ICHETEST parameter list must be in modifiable storage.
- If RELEASE=1.8 or later, the expansion of the ICHETEST macro MF=L or MF=I includes a one-byte field at offset 3 whose low-order bit is set to X'01' if the test failed because FLDACC=YES was specified on the associated ICHEINTY.
- It is possible to mix 1.7 and 1.8 or later format tests in the same request. The ICHEINTY and ICHEACTN macros can specify either RELEASE=1.7 or RELEASE=1.8 or later.
- When ICHEINTY LOCATE is used to retrieve data from a profile segment other than the BASE segment, default values (binary zeros for fixed-length fields, lengths of zero for variable length fields) are returned by the manager if the profile could contain but does not contain an occurrence of that segment. If you need to know whether the segment actually exists, specify a TEST for the SEGNAME on the ICHEINTY. For example, when doing a LOCATE to retrieve the TSO segment from a user profile, use TEST as follows:

```

    ICHETEST  FIELD=SEGNAME,COND=EQ,FLDATA=(8,CTSO)
    ....
    ....
    CTSO  DC  CL8'TSO'

```

ICHEACTN macro

You can use the ICHEACTN macro together with the ICHEINTY macro to retrieve or alter data in a specified RACF profile. ICHEACTN builds a parameter list containing the RACF profile field name and, optionally, the addresses of ICHETEST macros that control the data processing.

The ICHEACTN macro must be issued from a task running in non-cross-memory mode with no locks held. The issuing task must be authorized (APF-authorized, in system key 0-7, or running in supervisor state).

The format of the ICHEACTN macro is:

```

[label] ICHEACTN FIELD=field-name | address
              ,FLDATA=(length,address) | 'DEL' | 'COUNT'
              [,TESTS=(address[,AND,address]...)]
              [,RUN=YES | NO]
              [,GROUP=YES | NO]
              [,ENCRYPT=TEMPLATE | YES | NO]
              [,MF=L | (E,address) | I]
              [,RELEASE=number | (,CHECK) | (number,CHECK)]

```

FIELD=field-name | address

Specifies the field-name in the RACF profile whose value is to be retrieved or updated. The field must be one that is defined in the RACF database template.

Do not specify FIELD to be the first field in a database segment because the user cannot retrieve or update the first field in a segment. In the database templates, this field has a field ID of 001, and is usually described in the '**Field being described**' column as 'Start of segment fields'.

If you use the LIST form of the macro, specify the name of the field. The name must be 1 to 8 characters long, and not enclosed in quotation marks.

If you use the EXECUTE or INLINE form of the macro, specify the address of the name of the field to be retrieved or updated. The address can be an A-type address or register (2 through 12). For EXECUTE and INLINE, you can also specify the field name as a constant (for example, 'OWNER').

Do not alter a repeat group count field. Doing so causes unpredictable results and could corrupt the profile.

FLDATA=(length,address) | 'DEL' | 'COUNT'

Updates or deletes data in a specified RACF profile. This operand is valid when used with the ALTER, ALTERI, ADD, and RENAME operations on the ICHEINTY macro. It is also valid with LOCATE, NEXT, or

NEXTC if RELEASE=1.8 or later. The ICHEACTN macro has eight bytes reserved to hold the length and address of the retrieved data. In no case will a LOCATE, NEXT, or NEXTC return data into a field whose address is given in the ICHEACTN macro.

When you use ICHEACTN to replace modify data, the address points to a field that contains the value that is to replace the data in the specified FIELD of the profile. The address can be an A-type address or general register ((2) through (12)). The length specifies the size of the replacement field, and must be an integer constant or register ((2) through (12)).

When you use ICHEACTN to retrieve data and you specify RELEASE=1.8 or later, the RACF manager places the size of the retrieved field in the word at offset 12(X'0C') and the address of the data in the word at offset 16(X'10') of the ICHEACTN parameter list if no tests are specified. The addresses specified in TESTS= are placed before the FLDATA entries within the parameter list. Therefore, for each address noted within TESTS=, the FLDATA entries are displaced by four bytes. The use of the TESTS= operand increments these offsets by four bytes for each test specified regardless of whether DATAMAP=NEW or DATAMAP=OLD is specified.

'DEL' causes the field named in the FIELD operand to be given a null value or causes an occurrence of a repeat group to be deleted, or (if GROUP=YES is coded) deletes all occurrences of a repeat group.

If you are deleting an encrypted field, you must specify ENCRYPT=NO.

'COUNT' causes field-name in the FIELD operand to be treated as a positive integer and increased by one, unless the high-order bit is on, in which case, "COUNT" is reset to the value zero.

'COUNT' is intended for integer values only. Nor should 'COUNT' be used for repeat group count fields.

When replacing or adding data, the length and address are processed as follows:

- If DATAMAP=OLD is specified or defaulted on the ICHEINTY:

If the address is 0 or omitted, the specified field is given a null value (a variable-length field is set to a length of 0; a flag field is set to X'00'; other fixed-length fields are set to all 'FF').

If the length is 0 or omitted, and the address is specified, the result depends upon whether the specified field is a variable-length field or a fixed-length field.

- For a variable-length field, the field is given a null value. The length of the field is set to 0.
- For a fixed-length field or a flag field, the field is given the value pointed to by the specified address. The length of the field is taken from the template.

- If DATAMAP=NEW is specified on the ICHEINTY:

If the length is 0 or omitted, or the address is 0 or omitted, the field is given a null value as indicated above. Otherwise, the field is set from the data specified, with the length specified. For a fixed-length field, if the specified length is less than the length given in the template, the value is left-adjusted and filled with X'00's to the template length. If the length is greater than the template length, the operation fails. For variable-length fields, the specified length is used; the first byte of the data is not used as the data length, but rather is considered to be data.

TESTS= (address[,AND],address)...

Specifies preliminary testing that must occur before any data retrieval or updating takes place. Each address specified must be the address of a list built by an ICHETEST macro. The address can be an A-type address or register (2 through 12). Multiple addresses indicate that all conditions (tests) must be satisfied. If not, RACF suppresses further processing of the macro. If you omit the logical connector 'AND', you must use a comma to indicate its omission.

Note: If GROUP=YES is also coded on the ICHEACTN macro, all tests specified by the TESTS parameter are ignored unless RELEASE=1.8 or later is also specified.

The addresses specified in TESTS= are placed before the FLDATA entries within the parameter list. Therefore, for each address noted within TESTS=, the FLDATA entries are displaced by four bytes. The use of the TESTS= operand increments these offsets by four bytes for each test specified regardless of whether DATAMAP=NEW or DATAMAP=OLD is specified.

RUN= YES | NO

Specifies if a data retrieval or update is to be actually performed. This operand allows you to code an ACTION operand on the ICHEINTY macro without the action being performed for this particular execution. The default is RUN=YES.

GROUP= YES | NO

Specifies whether an update for a repeat group is for a single occurrence of the group or for the entire group, including the repeat count that contains the number of occurrences. If FIELD=field-name contains the name of a repeat group count field and GROUP=YES, ICHEACTN replaces or deletes the entire repeat group, including the count field. The data format used with GROUP=YES depends on the DATAMAP value on the ICHEINTY. See [“Using ICHEACTN to alter data when the ICHEINTY has DATAMAP=NEW”](#) on page 513 and [“Using ICHEACTN to alter data when ICHEINTY has DATAMAP=OLD”](#) on page 516 for details.

Note: If GROUP=YES is also coded on the ICHEACTN macro all tests specified by the TESTS parameter are ignored unless RELEASE=1.8 or later is specified.

ENCRYPT= TEMPLATE | YES | NO

Specifies whether the data specified by FLDATA is to be encoded. If ENCRYPT=YES, the data is encoded regardless of whether the template flag associated with the field specifies that it is to be encoded. If ENCRYPT=NO, RACF does not encode the data regardless of the template flag value. If ENCRYPT=TEMPLATE, the template flag determines whether the data is encoded.

MF= L | (E,address) | I

Specifies the form of the macro as either LIST, EXECUTE or INLINE.

The LIST form reserves and initializes storage. MF=L is the default. If RELEASE=1.8 or later is specified, the storage must be modifiable, that is, not within a re-entrant module.

The EXECUTE form modifies a list defined elsewhere. If you use the EXECUTE form, you must specify the address of the list to be modified. The address can be an A-type address or register (2 through 12).

The INLINE form is similar to a STANDARD form, except that it generates code to branch around the parameter list. In the MF=I form, the label names the first location of the parameter list, not the preceding instruction.

RELEASE=number**RELEASE=(,CHECK)****RELEASE=(number,CHECK)**

Specifies the release number. The release numbers that you can specify with the ICHEACTN macro are PLV0001, 77B0, 77A0, 7790, 7780, 7770, 7760, 7750, 7740, 7730, 7720, 7709, 7708, 7707, 7706, 7705, 7703, 2608, 2.6, 2.4, 2.3, 2.2, 2.1, 1.9.2, 1.9, 1.8.1, 1.8, or 1.7.

When you specify 1.8 or later, the RACF manager returns data using the 1.8 user work area format (documented in the topic [“Using ICHEACTN to retrieve data when ICHEINTY has DATAMAP=NEW”](#) on page 511). In effect, DATAMAP defaults to DATAMAP=NEW, if you specify RELEASE=1.8 or later and omit DATAMAP.

If you specify RELEASE=1.7 or allow the release parameter to default to 1.7, the RACF manager returns data using the 1.7 user work area format. In this case, DATAMAP defaults to DATAMAP=OLD if you omit it.

If you want to use 1.8 parameters, and the 1.7 user work area format, you must specify RELEASE=1.8 or later and DATAMAP=OLD.

To use the 1.8 parameters, you must specify RELEASE=1.8 or later. If you specify RELEASE=1.8 or later, the ICHEINTY parameter list must be in modifiable storage. The parameter list includes at offset 3 a byte whose low-order bit (X'01') is set if the action failed because of field level access checking.

The default is RELEASE=1.7.

Table 310. ICHEACTN parameters

Parameter	RELEASE=1.7 and earlier	RELEASE=1.8 or later
ENCRYPT=	X	X
FIELD=	X	X
FLDATA=	X	X
GROUP=	X	X
MF=	X	X
RUN=	X	X
TESTS=	X	X

Using ICHEACTN with the DATAMAP=NEW and DATAMAP=OLD operands

Installations can choose between using the old datamap format and the release 1.8 datamap format. The following sections explain the relationship between the DATAMAP keyword and the RELEASE keyword. In addition, this section explains how to use the ICHEACTN to retrieve and alter data when the ICHEINTY macro has DATAMAP=NEW specified and how to use ICHEACTN when the ICHEINTY macro has DATAMAP=OLD specified.

Using ICHEACTN to retrieve data when ICHEINTY has DATAMAP=NEW

The ICHEACTN macro retrieves data when used with the ICHEINTY macro that has a LOCATE, NEXT, or NEXTC operand. With DATAMAP=NEW on the ICHEINTY and RELEASE=1.8 or later on the ICHEACTN, data retrieval and modification are compatible operations. That is, you can do an ICHEINTY LOCATE followed by an ICHEINTY ALTER (with the same ICHEACTN) and the profile ends up with its original data. Or alternatively, by changing the ENTRY name you might copy data from one profile to another. When using ICHEACTN to retrieve data, you must supply a work area on the ICHEINTY macro into which the retrieved data can be placed. The first fullword of the work area must be the length of the work area (including the first fullword itself). The minimum work area is 30 bytes, even if no data is being retrieved.

The format of the user work area is as follows:

Offset (hex)	Length	Description
0	4	Length of entire work area
4	6	RBA return area
A	1	Flags
B	1	Reserved
C	4	Duplicate data set name count
10	8	Reserved
18	4	Length of data returned into work area
1C	variable	Field value return area

Ensure that the storage in the work area from +4 to +1E is initialized to binary zeros. If the area is not initialized, it can be difficult to determine if the information returned by the RACF manager is present.

If the profile located has a generic name, bit 0 (X'80') of the flag byte at offset (X'0A') is set to on.

An ICHEINTY macro can have several ICHEACTN macros associated with it. For each ICHEACTN macro, the RACF manager returns into the field value return area:

- A 4-byte length field. This length field contains the length of the retrieved data for that particular ICHEACTN macro. Note that this 4-byte length field does not contain its own length.
- The retrieved data from the RACF profile.
 - Simple variable-length fields are not preceded by an additional length byte as in the old format.
 - Within a combination field, each field is preceded by its respective four byte length field.
 - An alias field (combination field made up of only one field) does not have an extra length field.
 - Repeat group count fields are four bytes long, not two.
 - When replacing or retrieving an entire repeat group using (GROUP=YES), the repeat group count field does not precede the data.

When multiple ICHEACTNs are used, each returns data immediately following the data (if any) returned by the preceding ICHEACTN.

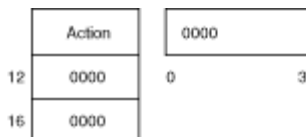
Note that all the fields are byte-aligned. In addition, if the ICHEACTN contains RELEASE=1.8 or later, the manager places the data length in the fullword at offset 12(X'0C') of the ICHEACTN and places a pointer to the data in the fullword at offset 16(X'10') of the ICHEACTN parameter list if no tests are specified. You must increment these offsets by 4 for each test specified by the ICHEACTN TESTS= parameter.

For example, with two tests, the length is returned at X'14' and the address is returned at X'18'. The addresses specified in TESTS= are placed before the FLDATA entries within the parameter list. Therefore, for each address noted within TESTS=, the FLDATA entries are displaced by four bytes.

The use of the TESTS= operand increments these offsets by four bytes for each test specified regardless of whether DATAMAP=NEW or DATAMAP=OLD is specified. The following examples show the format of the returned data (and the values that would be placed in the ICHEACTN if you specify RELEASE=1.8 or later).

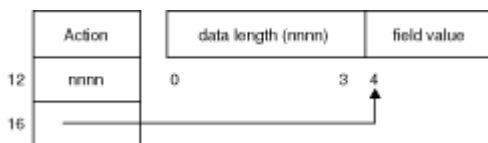
Some examples of the different field types that the RACF manager can return in the field value return area are:

1. If a condition specified by an ICHETEST macro (that is associated with the ICHEACTN macro) was not satisfied or if the specified field was a repeat field that contained no members, or if the action was failed by field level access checking, the field value area is not returned and the length area is equal to X'00000000'.



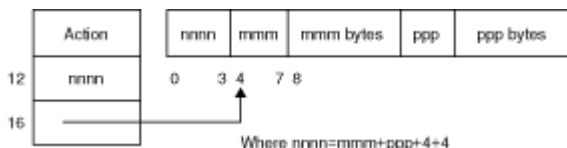
2. If the field specified is a fixed-length field, a variable-length field, a flag field, or a repeat group count field (GROUP=NO), the return field contains the length of the field followed by the field value.

Note: A flag field is always one byte long. A repeat group count field is always four bytes long if GROUP=NO. An alias field is processed the same way as the simple field of which it is an alias.



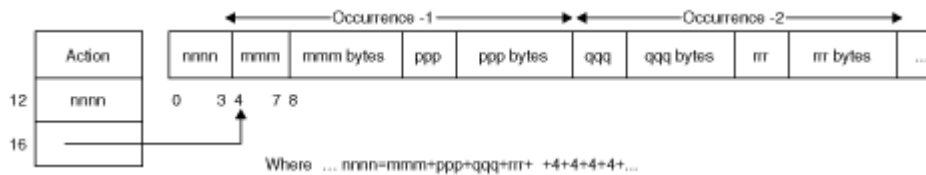
3. If the field specified is a combination field, the return area contains the length of all the fields in the combination, followed by a concatenation of the individual simple fields in the combination. Each simple field is returned as described above in (2).

For example, if the combination contains two simple fields:

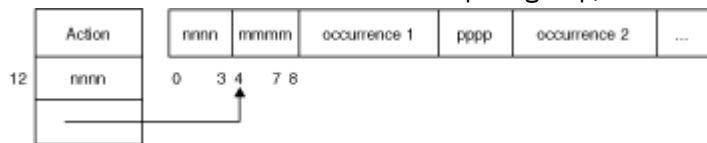


4. If the field specified is a field in a repeat group or a combination field made up of one or more fields in the same repeat group, the results returned depend upon whether
- An ICHETEST macro was associated with the ICHEACTN to position to a particular occurrence of the repeat group or
 - No ICHETEST macro was associated and all occurrences are implied.

When an ICHETEST is associated, the format of the result is the same as if the field were not in a repeat group. When no ICHETEST is associated, the result is the four-byte length field followed by the concatenation of the values of every occurrence of the specified field in the format shown above. If the specified field is a combination field, the values of the fields in the combination are first concatenated for each occurrence, then these concatenations are concatenated in the order of their occurrence.



5. If the field is a repeat-group count field, and the ICHEACTN specifies GROUP=YES, then the retrieved data contains all occurrences of the repeat group, in the following format:



Where nnnn is the total length of data returned, mmmm is the length of occurrence 1, and pppp is the length of occurrence 2.

Each occurrence is formatted as though it were a combination field (see example “3” on page 512) of all template fields defined for the group. For example, data set profiles have a field called ACLCNT; the fields in the group are USERID, USERACS, and ACSCNT. An ICHEACTN to retrieve ACLCNT, with GROUP=YES, would return the following data if ACLCNT has the value 2:

nnnn (length of data)	DC AL4(54)
mmmm (length of occurrence 1)	DC AL4(23)
Declares for occurrence 1	DC AL4(8)
	DC CL8 'userid1'
	DC AL4(1)
	DC AL1(useracs1)
	DC AL4(2)
	DC AL2(acscnt1)
pppp (length of occurrence 2)	DC AL4(23)
Declares for occurrence 2	DC AL4(8)
	DC CL8 'userid2'
	DC AL4(1)
	DC AL2(useracs2)
	DC AL4(2)
	DC AL2(acscnt2)

Using ICHEACTN to alter data when the ICHEINTY has DATAMAP=NEW

The ICHEACTN macro alters data when used with the ICHEINTY macro having an ADD, ALTER, ALTERI, or RENAME operand. If the conditions specified by the TESTS keyword on the ICHEACTN macro are met, the field specified in the FIELD operand is assigned the value specified in the FLDATA operand. If the specified field in the RACF profile is in a repeat group, then:

- If you specified a test with COND=EQ, the existing occurrence of the repeat group is altered.
- If you specified a test with COND=NE, a new occurrence is added to the end of the repeat group.
- If you did not specify a test, a new occurrence is added to the beginning of the repeat group.

When replacing data, the FLDATA parameter should describe the size of the data and its address in the same format as shown above for retrieving data. When specifying a combination field, the total size must equal the sum of the individual sizes, including the length fields or the request fails.

The specification of `FLDATA='COUNT'` causes the specified fields to be treated as a positive integer and increased by one. If the field specified is variable length or has a fixed length greater than four, RACF ignores the specification and does not modify the field value.

If you specify `FLDATA='DEL'`, the specified field has a null value; that is:

- For a fixed-length field that is not in a repeat group, the field is set to binary ones.
- For a flag field that is not in a repeat group, the field is set to binary zeros.
- For variable-length fields that are not in a repeat group, the length of the field is set to zero.
- For fields within a repeat group, the entire occurrence is deleted.

If you specify zero as the "address" value, the result is the same as if you had specified `FLDATA='DEL'`, except that for fields in a repeat group, the field in the occurrence is set to a null value (the same as fields not in a repeat group).

If you specify `FLDATA='DEL'` or `FLDATA='COUNT'` on an `ICHEACTN`, the length field of the `ICHEACTN` is set to -1 or -2. If you also specify `RELEASE=1.8` or later, and then use the `ICHEACTN` to retrieve data, these new values are lost. To avoid this, you should not use the same `ICHEACTN` for both `DEL/COUNT` and retrieval processing; or you should use the eForm to re-establish `DEL/COUNT` after the data retrieval.

Using ICHEACTN to retrieve data when the ICHEINTY has DATAMAP=OLD

The `ICHEACTN` macro retrieves data when used with the `ICHEINTY` macro having a `LOCATE`, `NEXT`, or `NEXTC` operand. When using `ICHEACTN` to retrieve data, you must supply a work area on the `ICHEINTY` macro into which the retrieved data can be placed. The first fullword of the work area must be the length of the work area (including the first fullword itself). The minimum work area is 30 bytes, even if no data is being retrieved.

The format of the user work area is as follows:

Offset (hex)	Length	Description
0	4	Length of entire work area
4	6	RBA return area
A	1	Flags
B	1	Reserved
C	4	Duplicate data set name count
10	8	Reserved
18	4	Length of data returned into work area
1C	variable	Field value return area

Ensure that the storage in the work area from +4 to +1E is initialized to binary zeros. If the area is not initialized, it can be difficult to determine if the information returned by the RACF manager is present.

If the profile located has a generic name, bit 0 (X'80') of the flag byte at offset (X'0A') is on. This flag bit is useful when performing `NEXT` or `NEXTC` operations to process many profiles.

An `ICHEINTY` macro can have several `ICHEACTN` macros associated with it. For each `ICHEACTN` macro, the RACF manager returns into the field value return area:

- A 2-byte length field. This length field contains the length of the retrieved data for that particular `ICHEACTN` macro. Note that this 2-byte length field does not contain its own length.
- The retrieved data from the RACF profile.

Note that all the fields are byte-aligned. In addition, if the `ICHEACTN` contains `RELEASE=1.8` or later, the manager places the data length in the fullword at offset 12 (X'0C') of the `ICHEACTN`, and places a pointer

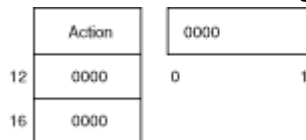
to the data in the fullword at offset 16(X'10') of the ICHEACTN parameter list if no tests are specified. You must increment these offsets by 4 for each test specified by the ICHEACTN TESTS= parameter.

For example, with two tests, the length is returned at X'14' and the address is returned at X'18'. The addresses specified in TESTS= are placed before the FLDATA entries within the parameter list. Therefore, for each address noted within TESTS=, the FLDATA entries are displaced by four bytes. The use of the TESTS= operand increments these offsets by four bytes for each test specified regardless of whether DATAMAP=NEW or DATAMAP=OLD is specified.

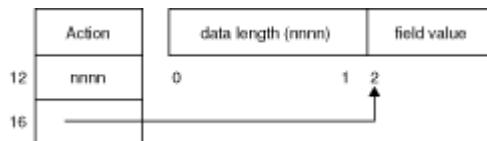
The following examples show the format of the returned data (and the values that would be placed in the ICHEACTN if you specify RELEASE=1.8 or later).

Some examples of the different field types that the RACF manager can return in the field value return area are:

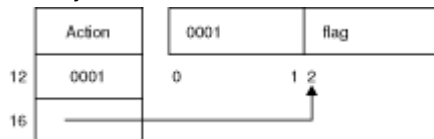
1. If a condition specified by an ICHETEST macro (that is associated with the ICHEACTN macro) was not satisfied or if the specified field was a repeat field that contained no members, the field value area is not returned and the length area is equal to X'0000'.



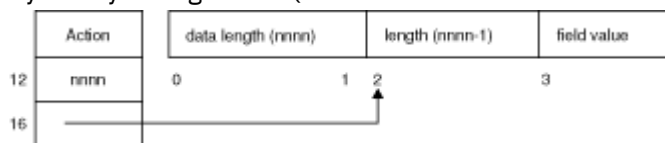
2. If the field specified is a fixed-length field, the return field contains the length of the field followed by the field value.



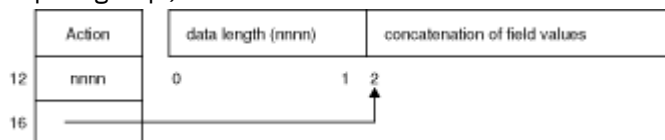
3. If the field specified is a flag field, the return field contains the length of the field (X'0001') followed by a 1-byte value.



4. If the field specified is a variable-length field, the return field contains the length of the field followed by a 1-byte length field (that does not include its own length) followed by the field value.

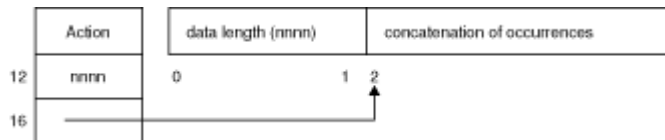


5. If the field specified is a combination field, the return area contains the length of all the fields in the combination, followed by a concatenation of values of each of the individual fields in the combination. If a field in the combination is in a repeat group, all the fields in the combination must be in the same repeat group. (Example 6 shows how the RACF manager returns combinations containing fields of a repeat group.)



6. If the field specified is a field in a repeat group or a combination field made up of one or more fields in the same repeat group, the results returned depend on whether (1) an ICHETEST macro was associated with the ICHEACTN to position to a particular occurrence of the repeat group or (2) no ICHETEST macro was associated and all occurrences are implied.

When an ICHETEST is associated, the format of the result is the same as if the field were not in a repeat group. When no ICHETEST is associated, the result is the two-byte length field followed by the concatenation of the values of every occurrence of the specified field. If the specified field is a combination field, the values of the fields in the combination are first concatenated for each occurrence, then these concatenations are concatenated in the order of their occurrence.



Using ICHEACTN to alter data when ICHEINTY has DATAMAP=OLD

The ICHEACTN macro alters data when used with the ICHEINTY macro having an ADD, ALTER, ALTERI, or RENAME operand. If the conditions specified by the TESTS keyword on the ICHEACTN macro are met, the field specified in the FIELD operand is assigned the value specified in the FLDATA operand. If the specified field in the RACF profile is in a repeat group, then:

- If you specified a test with COND=EQ, the existing occurrence of the repeat group is altered.
- If you specified a test with COND=NE, a new occurrence is added to the end of the repeat group.
- If you did not specify a test, a new occurrence is added to the beginning of the repeat group.

RACF uses the length specified as a subfield of the FLDATA keyword only when you specify GROUP=YES. For fixed-length fields, the data length is the field length in the template. For variable-length fields, the data length is the first data byte (it does not include its own length). RACF handles combination fields as a succession of fields, either fixed or variable length. If the combination field contains some but not all of the fields in a repeat group, the fields not included are set to null values.

The specification of FLDATA='COUNT' causes the specified field to be treated as a positive integer and increased by one. If the field specified is variable length or has a fixed length greater than four, RACF ignores the specification and does not modify the field value.

If you specify FLDATA='DEL', the specified field is given a null value; that is:

- For a fixed-length field that is not in a repeat group, the field is set to binary ones.
- For a flag field that is not in a repeat group, the field is set to binary zeros.
- For variable-length fields that are not in a repeat group, the length of the field is set to zero.
- For fields within a repeat group, the entire occurrence is deleted.

If you specify zero as the "address" value, the result is the same as if you had specified FLDATA='DEL', except that for fields in a repeat group, the field in the occurrence is set to a null value (the same as fields not in a repeat group).

Examples of ICHEINTY, ICHETEST, and ICHEACTN macro usage

The following examples illustrate some of the functions provided by the ICHEINTY, ICHETEST, and ICHEACTN macros:

Example 1. Determining whether a user is defined to RACF:

```
*      .
*      .
*      .
*      LA    15,WEND-W      LENGTH OF WORK AREA.
*      ST    15,W           INITIALIZE WORK AREA.
*      XC    WR,WR          CLEAR RESERVED AREA.
*      ICHEINTY LOCATE,TYPE='USR',ENTRY=USR1,WKAREA=W
*      LTR    15,15         R15=0 IF USER DEFINED TO
*                           RACF
*      BNZ    NOTDEFD
*      .
*      .
*      .
```

```

*      DATA AREAS
USR1   DS    AL1      LENGTH OF USERID (1 TO 8)
        DS    CL8      USERID
W       DS    0F
        DS    F        LENGTH OF WORK AREA.
WR      DS    CL24     RESERVED.
        DS    F
WEND    EQU    *      END OF WORK AREA.

```

The ICHEINTY macro identifies the user profile to be located. A return code of 0 (X'00') in register 15 indicates that the user is defined to RACF. A return code of 12 (X'0C') indicates that the user is not defined. Note that this ICHEINTY macro contains a work area. By also coding an ICHEACTN macro in this example, you can retrieve current field values from this user profile into the work area.

Example 2. Adding a user ID to a data set access list:

```

*      .
*      .
*      .
*      ICHEINTY ALTER,TYPE='DS',ENTRY=DSN1,          *
*      ACTIONS=AACL
*      LTR    15,15      0 RETURNED IF DS IS RACF
*                      DEFINED
*      BNZ    DSNOTDEF   DS NOT RACF DEFINED OR
*                      ERROR
*      CLI    TUSERID+1,X'00' WAS USER ALREADY IN LIST
*      BNZ    INLIST     YES.  USER WAS IN LIST
*                      ALREADY
*      .
*      .
*      .
*      DATA AREA
AACL    ICHEACTN FIELD=ACL,FLDATA=(11,ACL),          *
        TESTS=TUSERID,MF=L
TUSERID ICHETEST FIELD=USERID,FLDATA=(8,USER),COND=NE, *
        MF=L
DSN1    DS    AL1      DATA SET NAME LENGTH
        DS    CL44     (1 TO 44)
ACL     DS    0CL11    DATA SET NAME
USER    DS    CL8      ACCESS LIST ENTRY
USERACS DS    XL1      USERID TO BE ADDED
*                      ACCESS TO BE GIVEN:
*                      X'80' FOR ALTER
*                      X'40' FOR CONTROL
*                      X'20' FOR UPDATE
*                      X'10' FOR READ
*                      X'01' FOR NONE
ACSCNT  DC    XL2'0000' ZERO ACCESS COUNT

```

The ICHEINTY macro identifies the data set profile whose access list is to be updated. It also points to an ICHEACTN macro that describes how the profile is to be updated. In this example, RACF adds a user ID to the access list.

The ICHEACTN macro, in turn, points to an ICHETEST macro that tests for certain conditions before the profile can be updated. In this example, ICHETEST tests to determine if the specified user ID already exists in the access list. (The second byte of the test block at TUSERID is 0 if the user ID is not in the access list.) If the user ID does not exist, RACF adds the user ID (with the specified access authority) to the access list and updates the data set profile. If the user ID already exists, no profile update occurs.

Example 3. Changing the access authority of a user in a data set access list:

```

*      .
*      .
*      .
*      ICHEINTY ALTER,TYPE='DS',ENTRY=DSN1,          *
*      ACTIONS=AUSRACS
*      LTR    15,15      0 RETURNED IF DS IS RACF
*                      DEFINED
*      BNZ    DSNOTDEF   DS NOT RACF DEFINED OR
*                      ERROR
*      CLI    TUSERID+1,X'00' WAS USER IN LIST
*      BNZ    NOTINLST   NO.  USER WAS NOT IN
*                      LIST
*      .
*      .

```

Examples

```

*
*      DATA AREA
AUSRACS  ICHEACTN FIELD=USERACS,FLDATA=(1,USERACS),      *
          TESTS=TUSERID,MF=L
TUSERID  ICHEACTN FIELD=USERID,FLDATA=(8,USER),COND=EQ,  *
          MF=L
DSN1     DS    AL1          DATA SET NAME LENGTH
          (1 TO 44)
          DATA SET NAME
UACC     DS    CL44          ACCESS TO BE GIVEN:
*         X'80' FOR ALTER
*         X'40' FOR CONTROL
          X'20' FOR UPDATE
          X'10' FOR READ
          X'01' FOR NONE

```

This example is similar to the previous example. However, if the user ID exists in the data set access list, RACF changes that user's access authority to the value specified in USERACS and updates the data set profile. If the user ID does not exist, no profile update occurs.

Note that you can use this example to delete a user ID from the data set access list by changing the ICHEACTN macro to read:

```

AUSRACS  ICHEACTN FIELD=USERID,FLDATA='DEL',            *
          TEST=TUSERID,MF=L

```

Example 4. Retrieving owner names of all data set profiles:

The following example program shows an ICHEINTY coded to retrieve the owner names of all data set profiles in the RACF database.

```

EXAMPLE  CSECT
*
*      entry linkage
*
      STM 14,12,12(13)          push caller registers
      BALR 12,0                establish ...
      USING *,12                ... addressability
      GETMAIN R,LV=DYNLEN       get dynamic storage
      LR 11,1                  move getmained address to R11
      USING DYNAREA,11          addressability to DSECT
      ST 13,SAVEAREA+4          save caller save area address
      LA 15,SAVEAREA            get address of own save area
      ST 15,8(13)               store in caller save area
      LR 13,15                  get address of own save area
*
*      initialize variables in dynamic storage area
*
      MVC ENTBLN,H44            set buffer length to 44
      MVC ENTNLEN,H1            set entity length to 1
      XC ENTNAME,ENTNAME        clear entity name area
      MVC RETALEN,F40           set return area length
*
*      copy static ICHEINTY and ICHEACTN to dynamic GETMAINED areas
*
      MVC DYNICH(ICHLEN),STATIC
      MVC DYNACT(ACTLEN),STACT
      ICHEINTY RELEASE=1.9,ACTIONS=(DYNACT),WKAREA=RETAREA,      *
          OPTIONS=(FLDEF,NOEXEC),GENERIC=NO,MF=(E,DYNICH)
*
*      loop to retrieve all data set profiles
*      for each high level qualifier, generic profiles are
*      retrieved first
*
      LOOP EQU *                start of loop
      XC RETDATA,RETDATA        clear ICHEINTY return data
      ICHEINTY NEXTC,ENTRYX=ENTBUFF,RELEASE=1.9,MF=(E,DYNICH)
      LTR 15,15                 check return code
      BNZ DONE                  exit on non zero return code
*
*      .
*      .
*      process data set profiles
*      .
*      .
      TM RETFLAGS,X'80'         check generic bit
      BO GENERIC                branch if generic bit is on
      ICHEINTY OPTIONS=(NOEXEC),GENERIC=NO,MF=(E,DYNICH)

```

```

B      LOOP                                process next profile
*
GENERIC EQU   *                            profile name is generic
        ICHEINTY OPTIONS=(NOEXEC),GENERIC=UNCOND,MF=(E,DYNICH)
        B      LOOP                        process next profile
*
*      return to caller
*
DONE    EQU   *                            return to caller
        L      13,SAVEAREA+4              caller's save area address
        FREEMAIN R,LV=DYNLEN,A=(11)       free dynamic storage
        LM     14,12,12(13)               pop registers
        SLR    15,15                       clear return code
        BR     14                          return to caller
*
*      static ICHEACTN and ICHEINTY areas
*
STATACT ICHEACTN FIELD=OWNER
ACTLEN  EQU   *-STATACT                    length of ICHEACTN
*
STATICH ICHEINTY NEXTC,TYPE='DS',ENTRYX=-*,RELEASE=1.9,DATAMAP=NEW, *
        ACTIONS=(STATACT),WKAREA=-*,MF=L
ICHLLEN EQU   *-STATICH                    length of ICHEINTY
*
*      constants
*
H1      DC    H'1'
H44     DC    H'44'
F40     DC    F'40'
*
*      dynamic area
*
DYNAREA DSECT
*
SAVEAREA DC    18F'0'
DYNICH   DS    17F                           dynamic ICHEINTY area
DYNACT   DS    6F                           dynamic ICHEACTN area
*
*      ENTITYX structure
*
ENTBUFF  DS    0CL48
ENTBLEN  DS    H
ENTNLEN  DS    H
ENTNAME  DS    CL44
*
*      return work area
*
RETAREA  DS    0CL40
RETALLEN DS    F                             return area length
RETDATA  DS    0CL36
RETRBA   DS    CL6                           RBA return area
RETFLAGS DS    CL1                           flags
RETRES1  DS    CL1                           reserved
RETDDSC  DS    F                             duplicate data set name count
RETRES2  DS    CL8                           reserved
RETDLEN  DS    F                             returned data length
RETOWNLN DS    F                             returned owner name length
RETOWNER DS    CL8                           returned owner name
*
DYNLEN   EQU   *-DYNAREA                     dynamic area length
*
END
```

Example 5. Updating the installation fields:

The RACF template defines a repeat group of fields for installation use. There are four of these fields:

USRCNT

Contains the number of repeat members in the group. A repeat member is one USRNM field, one USRDATA field, and one USRFLAG field.

USRNM

Describes the contents of the USRDATA field.

USRDATA

Contains any information that you choose.

USRFLAG

Is a flag associated with USRNM.

The following example shows how the installation fields are used:

```
USRCNT = 2
      USRNM  ACCTNMBR
      USRDATA K83-1234/DQ3
      USRFLG  00

      USRNM  ADDRESS
      USRDATA RFD 4, Box 7711, Phoenicia, NY
      USRFLG  00
```

The following example shows how to add or update a repeat group member. This code will first delete an existing occurrence, based on the name in USRNM, and then add a new occurrence with the wanted new (or updated) data. The code is assumed to be preceded by code that initializes the UDATANM, UDATAL1 and UDATAV fields.

In the part of the example that is not shown, the ACTN3 and ACTN4 macros are addressed by an ICHEINTY-ALTER macro. The ACTN3 and ACTN4 macros must be specified in the ICHEINTY-ACTIONS keyword in the order ACTN3, ACTN4.

```
      ICHEACTN MF=(E,ACTN3),TESTS=TEST3
      ICHETEST MF=(E,TEST3),FLDATA=(,UDATANM)
      ICHEACTN MF=(E,ACTN4),FLDATA=((Rx),UDATA),TESTS=TEST4
      ICHETEST MF=(E,TEST4),FLDATA=(,UDATANM)
      .
      .
      .
      --- Invoke ICHEINTY ---
      .
      .
      .
ACTN3  ICHEACTN FIELD=USRNM,FLDATA='DEL',TESTS=**-*
TEST3  ICHETEST FIELD=USRNM,FLDATA=(8,**-*)          COND=EQ is default.
ACTN4  ICHEACTN FIELD=USERDATA,FLDATA=(**-*,**-*),TESTS=**-*

TEST4  ICHETEST FIELD=USRNM,FLDATA=(8,**-*),COND=NE

UDATA  DS      0C          Start of USERDATA area.
UDATANM DS      CL8        Contents of USRNM field.
UDATAL1 DS      AL1        Length of USRDATA field.
UDATAV  DS      CL--       Contents of USRDATA field.
*
* The USRFLG field will be at an offset of UDATAL1+1 from
* the beginning of the UDATAV field.
*
```

Appendix B. REXX RACVAR

The REXX RACVAR function is a RACF service for REXX execs; it provides information about the running user.

The REXX RACVAR function has four arguments. It provides information about:

USERID

The user ID that is in the ACEE

GROUPID

The group name that is in the ACEE

SECLABEL

The security label that is in the ACEE

ACEESTAT

The status of the ACEE. The function returns NO ACEE, DEFAULT, DEFINED, or UNDEFINED

Below is a sample REXX exec that uses RACVAR to check the USERID, GROUPID, and SECLABEL in the user's ACEE.

```
/* rexx */
say "Current ACEE status is " racvar('ACEESTAT') "."
if racvar('ACEESTAT') = 'NO ACEE' then
do
  say ' You have no ACEE defined'
end
else
do
  say "Your user ID is " racvar('USERID') "."
  say "You are connected to group " racvar('GROUPID') "."
  current_seclabel = racvar('SECLABEL')
  if current_seclabel = ' ' then
  do
    say ' You have no SECLABEL defined'
  end
  else
  do
    say "Your SECLABEL is " current_seclabel "."
  end
end
end
return
```

To execute the REXX RACVAR function, your REXX parameter module must contain an entry for RACF's IRRFPCK directory package which, in turn, supports the RACVAR function. For descriptions of REXX parameter modules and updating and integrating them, see *z/OS TSO/E REXX Reference* in the topics that describe Programming Services, Function Packages, and function directories.

Appendix C. Supplied class descriptor table entries

This appendix contains a table that describes the IBM-supplied class entries and attributes in the class descriptor table (ICHRRCDX).

- “Supplied class descriptor table entries” on page 523

Supplied class descriptor table entries

Table 311 on page 523 lists the class entries supplied by IBM in the class descriptor table (ICHRRCDX). Other classes can be added to the class descriptor table (CDT) by your installation.

Programming interface information: The class descriptor table (ICHRRCDX) is mapped by the ICHPCNST macro. Programming interface information for ICHPCNST is found in the data area section called CSNT/CSNX (RACF) in *z/OS Security Server RACF Data Areas* in the *z/OS Internet library* (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary).

Table 311. Classes supplied by IBM		
Class	Attributes	
ACCTNUM	POSIT=126	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=39
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=46
	FIRST=ANY	
ACECHK	POSIT=605	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
		SLBLREQ=NO
		RVRSMAC=NO
	OPER=NO	KEYQUAL=0
	PROFDEF=YES	ID=1
	FIRST=ANY	CASE=UPPER
	SIGNAL=YES	
		GENERIC=ALLOWED

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
ACICSPCT	POSIT=5	OTHER=ANY
		MAXLNTH=13
		DFTRETC=4
		DFTUACC=NONE
	GROUP=BCICSPCT	
	OPER=NO	
		ID=37
	FIRST=ANY	
AIMS	POSIT=4	OTHER=ALPHANUM
		MAXLNTH=8
		DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=11
	FIRST=ALPHA	
ALCSAUTH	POSIT=548	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=62
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
APPCLU	POSIT=118	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=35
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=57
	FIRST=ALPHA	
APPCPORT	POSIT=87	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
		RVRSMAC=YES
	OPER=NO	
	PROFDEF=YES	ID=98
	FIRST=ALPHA	MAXLENX=17
APPCSERV	POSIT=84	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=73
	GENLIST=DISALLOWED	DFTRETC=8
	RACLREQ=YES	DFTUACC=NONE
		SLBLREQ=YES
	OPER=NO	
	PROFDEF=YES	ID=105
	FIRST=ALPHANUM	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
APPCSI	POSIT=88	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=26
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=READ
	OPER=NO	
	PROFDEF=YES	ID=97
	FIRST=ALPHANUM	
APPCTP	POSIT=89	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=82
	GENLIST=DISALLOWED	DFTRETC=8
	RACLREQ=YES	DFTUACC=NONE
		SLBLREQ=YES
	OPER=NO	
	PROFDEF=YES	ID=96
	FIRST=ALPHANUM	
APPL	POSIT=3	OTHER=ALPHANUM
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=ALLOWED	DFTRETC=4
		DFTUACC=NONE
		SLBLREQ=YES
		EQUALMAC=YES
	OPER=NO	
		ID=8
	FIRST=ALPHA	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
BCICSPCT	POSIT=5	OTHER=ANY
		MAXLNTH=13
		DFTRETC=4
		DFTUACC=NONE
	MEMBER=ACICSPCT	
	OPER=NO	
		ID=38
	FIRST=ANY	
CACHECLS	POSIT=569	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=16
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=NO
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ALPHANUM	
CBIND	POSIT=545	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=41
		DFTRETC=8
	OPER=NO	
		ID=1
	FIRST=ALPHA	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
CCICSCMD	POSIT=5	OTHER=ANY
		MAXLNTH=21
		DFTRETC=4
		DFTUACC=NONE
	GROUP=VCICSCMD	
	OPER=NO	
		ID=52
	FIRST=ANY	
CDT	POSIT=572	OTHER=ALPHANUM
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=NO
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ALPHANUM	
CFIELD	POSIT=588	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=26
	GENLIST=DISALLOWED	
	RACLREQ=NO	DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ALPHA	
	SIGNAL=NO	GENERIC=DISALLOWED

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
CIMS	POSIT=93	OTHER=ALPHANUM
	RACLIST=DISALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	GROUP=DIMS	
	OPER=NO	
		ID=88
	FIRST=ALPHA	
CONSOLE	POSIT=107	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
		RVRSMAC=YES
	OPER=NO	
		ID=68
	FIRST=ANY	
CPSMOBJ	POSIT=57	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=44
	GENLIST=ALLOWED	DFTRETC=4
	GROUP=GCPSMOBJ	
	OPER=NO	
		ID=1
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
CPSMXMP	POSIT=11	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=44
		DFTRETC=4
	OPER=NO	
		ID=1
	FIRST=ANY	
CRYPTOZ	POSIT=578	OTHE=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	OPER=NO	KEYQUAL=0
		ID=1
	FIRST=ANY	
CSFKEYS	POSIT=98	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=73
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	GROUP=GCSFKEYS	
	OPER=NO	
		ID=100
	FIRST=ALPHA	MAXLENX=246
	SIGNAL=YES	

Table 311. Classes supplied by IBM (continued)

Class	Attributes	
CSFSERV	POSIT=98	
	RACLIST=ALLOWED	OTHER=ANY
	GENLIST=DISALLOWED	MAXLNTH=8
	RACLREQ=YES	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=99
	FIRST=ALPHA	
	SIGNAL=YES	MAXLENX=246
DASDVOL	POSIT=0	OTHER=ALPHANUM
	RACLIST=ALLOWED	MAXLNTH=6
	GENLIST=ALLOWED	DFTRETC=4
	GROUP=GDASDVOL	
	OPER=YES	
		ID=5
DBNFORM	FIRST=ALPHANUM	
	POSIT=59	OTHER=ALPHANUM
	RACLIST=ALLOWED	MAXLNTH=8
		DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ALPHANUM	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
DCEUUIDS	POSIT=544	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=73
	GENLIST=ALLOWED	DFTRETC=8
		DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ALPHANUM	
DCICSDCT	POSIT=5	OTHER=ANY
		MAXLNTH=13
		DFTRETC=4
		DFTUACC=NONE
	GROUP=ECICSDCT	
	OPER=NO	
		ID=31
	FIRST=ANY	
DEVICES	POSIT=115	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=39
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
		SLBLREQ=YES
	OPER=NO	
		ID=60
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
DIGTCERT	POSIT=550	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	OPER=NO	KEYQUAL=0
		ID=1
	FIRST=ANY	
DIGTCRIT	POSIT=563	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	OPER=NO	KEYQUAL=0
		ID=1
	FIRST=ANY	
DIGTNMAP	POSIT=563	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	OPER=NO	KEYQUAL=0
		ID=1
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
DIGTRNG	POSIT=550	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	OPER=NO	KEYQUAL=0
		ID=1
	FIRST=ANY	
DIMS	POSIT=93	OTHER=ALPHANUM
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	MEMBER=CIMS	
	OPER=NO	
		ID=89
DIRACC	POSIT=71	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
	OPER=NO	
	PROFDEF=NO	ID=107
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)

Class	Attributes	
DIRAUTH	POSIT=105	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
	OPER=NO	
	PROFDEF=NO	ID=70
	FIRST=ANY	
DIRECTRY	POSIT=95	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=153
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
		SLBLREQ=YES
	OPER=YES	KEYQUAL=2
		ID=86
DIRSRCH	POSIT=70	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
	OPER=NO	
	PROFDEF=NO	ID=106
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
DLFCLASS	POSIT=92	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=64
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=90
	FIRST=ALPHA	
DSNADM	POSIT=539	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	SIGNAL=YES	MAXLENX=246
DSNR	POSIT=7	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=39
	GENLIST=ALLOWED	
		SLBLREQ=YES
		EQUALMAC=YES
	OPER=NO	
		ID=18
	FIRST=ALPHANUM	

Table 311. Classes supplied by IBM (continued)

Class	Attributes	
DSNRAUTH	POSIT=610	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=8
	RACLREQ=YES	DFTUACC=NONE
	CLASS=DSNRAUTH	SLBLREQ=NO
		RVRSMAC=NO and EQUALMAC=NO
	OPER=NO	KEYQUAL=0
	PROFDEF=YES	ID=18
	FIRST=ANY	CASE=ASIS
	SIGNAL=YES	
		GENERIC=ALLOWED
ECICSDCT	POSIT=5	OTHER=ANY
		MAXLNTH=13
		DFTRETC=4
		DFTUACC=NONE
	MEMBER=DCICSDCT	
	OPER=NO	
		ID=32
	FIRST=ANY	
EJBROLE	POSIT=568	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GEJBROLE	SLBLREQ=NO
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	CASE=ASIS

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
FACILITY	POSIT=8	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=39
	GENLIST=ALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=19
	FIRST=ANY	
FCICSFCT	POSIT=5	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=17
		DFTRETC=4
		DFTUACC=NONE
	GROUP=HCICSFCT	
	OPER=NO	
		ID=27
	FIRST=ANY	
FIELD	POSIT=121	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=26
	GENLIST=ALLOWED	DFTRETC=4
		DFTUACC=NONE
	RACLREQ=YES	
	OPER=NO	
		ID=51
	FIRST=ALPHA	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
FILE	POSIT=94	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=171
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
		SLBLREQ=YES
	OPER=YES	KEYQUAL=2
		ID=87
	FIRST=ANY	
FIMS	POSIT=101	OTHER=ALPHANUM
	RACLIST=DISALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	GROUP=HIMS	
	OPER=NO	
		ID=79
	FIRST=ALPHANUM	
FSACCESS	POSIT=595	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=44
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
		SLBLREQ=NO
	OPER=NO	KEYQUAL=0
	PROFDEF=YES	ID=1
	FIRST=ANY	CASE=UPPER
	SIGNAL=YES	GENERIC=ALLOWED

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
FSEXEC	POSIT=599	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=44
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
		SLBLREQ=NO
	OPER=NO	KEYQUAL=0
	PROFDEF=YES	ID=1
	FIRST=ANY	CASE=ASIS
	SIGNAL=YES	GENERIC=ALLOWED
FSOBJ	POSIT=72	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
	OPER=NO	
	PROFDEF=NO	ID=108
	FIRST=ANY	
FSSEC	POSIT=73	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
	OPER=NO	
	PROFDEF=NO	ID=109
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
GCICSTRN	POSIT=5	OTHER=ANY
		MAXLNTH=13
		DFTRETC=4
		DFTUACC=NONE
	MEMBER=TCICSTRN	
	OPER=NO	
		ID=13
	FIRST=ANY	
GCPSMOBJ	POSIT=57	OTHER=ANY
		MAXLNTH=246
		DFTRETC=4
	MEMBER=CPSMOBJ	
	OPER=NO	
		ID=1
	FIRST=ANY	
GCSFKEYS	POSIT=98	OTHER=NONATNUM
	RACLIST=DISALLOWED	MAXLNTH=17
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	MEMBER=CSFKEYS	
	OPER=NO	
		ID=101
	FIRST=NONATABC	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
GDASDVOL	POSIT=0	OTHER=ALPHANUM
		MAXLNTH=6
		DFTRETC=4
	MEMBER=DASDVOL	
	OPER=YES	
		ID=39
	FIRST=ALPHANUM	
GDSNBP	POSIT=536	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
	MEMBER=MDSNBP	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
GDSNCL	POSIT=538	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
	MEMBER=MDSNCL	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
		MAXLENX=246

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
GDSNDB	POSIT=528	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
	MEMBER=MDSNDB	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
GDSNGV	POSIT=596	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
	MEMBER=MDSNGV	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
		MAXLENX=246
	SIGNAL=YES	
GDSNJR	POSIT=567	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
	MEMBER=MDSNJR	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
		MAXLENX=246

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
GDSNPK	POSIT=534	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=NO
	MEMBER=MDSNPK	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	SIGNAL=YES	
GDSNPN	POSIT=533	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
	MEMBER=MDSNPN	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	SIGNAL=YES	
GDSNSC	POSIT=562	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
	MEMBER=MDSNSC	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
		MAXLENX=246

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
GDSNSG	POSIT=537	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
	MEMBER=MDSNSG	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
		MAXLENX=246
GDSNSM	POSIT=535	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
	MEMBER=MDSNSM	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	SIGNAL=YES	
GDSNSP	POSIT=561	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
	MEMBER=MDSNSP	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	SIGNAL=YES	MAXLENX=246

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
GDSNSQ	POSIT=573	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=NO
	MEMBER=MDSNSQ	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	SIGNAL=YES	MAXLENX=246
GDSNTB	POSIT=530	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
	MEMBER=MDSNTB	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	SIGNAL=YES	MAXLENX=246
GDSNTS	POSIT=529	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
	MEMBER=MDSNTS	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
GDSNUF	POSIT=560	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
	MEMBER=MDSNUF	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	SIGNAL=YES	MAXLENX=246
GDSNUT	POSIT=559	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=NO
	MEMBER=MDSNUT	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
		MAXLENX=246
GEJBROLE	POSIT=568	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=NO
	MEMBER=EJBROLE	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	CASE=ASIS

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
GIMS	POSIT=4	OTHER=ALPHANUM
		MAXLNTH=8
		DFTRETC=4
		DFTUACC=NONE
	MEMBER=TIMS	
	OPER=NO	
		ID=10
	FIRST=ALPHA	
GINFOMAN	POSIT=85	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=44
	GENLIST=DISALLOWED	DFTRETC=4
	MEMBER=INFOMAN	
	OPER=NO	
		ID=104
	FIRST=ANY	
GLOBAL	POSIT=6	OTHER=ANY
		MAXLNTH=8
		DFTRETC=4
		DFTUACC=NONE
	MEMBER=GMBR	
	OPER=NO	
		ID=17
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
GMBR	POSIT=6	OTHER=ANY
		MAXLNTH=39
		DFTRETC=4
		DFTUACC=NONE
	GROUP=GLOBAL	
	OPER=NO	
		ID=16
	FIRST=ANY	
GMQADMIN	POSIT=80	OTHER=ANY
		MAXLNTH=62
		DFTRETC=8
		DFTUACC=NONE
	MEMBER=MQADMIN	
	OPER=NO	
		ID=121
	FIRST=ANY	
GMQCHAN	POSIT=58	OTHER=ANY
		MAXLNTH=53
		DFTRETC=8
		DFTUACC=NONE
	MEMBER=MQCHAN	
	OPER=NO	
		ID=1
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
GMQNLIST	POSIT=79	OTHER=ANY
		MAXLNTH=53
		DFTRETC=8
		DFTUACC=NONE
	MEMBER=MQNLIST	
	OPER=NO	
		ID=119
	FIRST=ANY	
GMQPROC	POSIT=78	OTHER=ANY
		MAXLNTH=53
		DFTRETC=8
		DFTUACC=NONE
	MEMBER=MQPROC	
	OPER=NO	
		ID=117
	FIRST=ANY	
GMQQUEUE	POSIT=77	OTHER=ANY
		MAXLNTH=53
		DFTRETC=8
		DFTUACC=NONE
	MEMBER=MQQUEUE	
	OPER=NO	
		ID=4
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
GMXADMIN	POSIT=586	OTHER=ANY
		MAXLNTH=62
		DFTRETC=8
		DFTUACC=NONE
	MEMBER=MXADMIN	
	OPER=NO	
		ID=1
	FIRST=ANY	CASE=ASIS
GMXNLIST	POSIT=585	OTHER=ANY
		MAXLNTH=53
		DFTRETC=8
		DFTUACC=NONE
	MEMBER=MXNLIST	
	OPER=NO	
		ID=1
	FIRST=ANY	CASE=ASIS
GMXPROC	POSIT=584	OTHER=ANY
		MAXLNTH=53
		DFTRETC=8
		DFTUACC=NONE
	MEMBER=MXPROC	
	OPER=NO	
		ID=1
	FIRST=ANY	CASE=ASIS

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
GMXQUEUE	POSIT=583	OTHER=ANY
		MAXLNTH=53
		DFTRETC=8
		DFTUACC=NONE
	MEMBER=MXQUEUE	
	OPER=NO	
		ID=1
	FIRST=ANY	CASE=ASIS
GMXTOPIC	POSIT=587	OTHER=ANY
		MAXLNTH=246
		DFTRETC=8
		DFTUACC=NONE
	MEMBER=MXTOPIC	
	OPER=NO	
		ID=1
	FIRST=ANY	CASE=ASIS
GSDSF	POSIT=100	OTHER=ANY
		MAXLNTH=63
		DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	MEMBER=SDSF	
	OPER=NO	
		ID=95
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
GSOMDOBJ	POSIT=547	OTHER=ANY
		MAXLNTH=246
		DFTRETC=8
	MEMBER=SOMDOBJS	
	OPER=NO	
		ID=1
	FIRST=ALPHA	
GTERMINL	POSIT=2	OTHER=ALPHANUM
		MAXLNTH=8
		DFTRETC=4
	MEMBER=TERMINAL	
	OPER=NO	
		ID=42
	FIRST=ALPHANUM	
GXCSFKEY	POSIT=98	OTHER=NONATNUM
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	
		DFTUACC=NONE
	MEMBER=XCSFKEY	
	OPER=NO	
		ID=101
	FIRST=NONATABC	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
GXFACILI	POSIT=8	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=ALLOWED	DFTRETC=4
		DFTUACC=NONE
	MEMBER=XFACILIT	
	OPER=NO	
		ID=1
	FIRST=ALPHANUM	
GZMFAPLA	POSIT=592	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=NO
	MEMBER=ZMFAPLA	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	CASE=ASIS
HBRADMIN	POSIT=603	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=64
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	
		SLBLREQ=NO
		EQUALMAC=NO
	OPER=NO	
	PROFDEF=YES	
	FIRST=ALPHANUM	CASE=UPPER
	SIGNAL=NO	GENERIC=ALLOWED

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
HBRCONN	POSIT=603	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=64
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	
		SLBLREQ=NO
		EQUALMAC=NO
	OPER=NO	
	PROFDEF=YES	
	FIRST=ALPHANUM	CASE=UPPER
	SIGNAL=NO	GENERIC=ALLOWED
HBRCMD	POSIT=603	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=64
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	
		SLBLREQ=NO
		EQUALMAC=NO
	OPER=NO	
	PROFDEF=YES	
	FIRST=ALPHANUM	CASE=UPPER
	SIGNAL=NO	GENERIC=ALLOWED
HCICSFCT	POSIT=5	OTHER=ANY
		MAXLNTH=17
		DFTRETC=4
		DFTUACC=NONE
	MEMBER=FCICSFCT	
	OPER=NO	
		ID=28
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
HIMS	POSIT=101	OTHER=ALPHANUM
	RACLIST=DISALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	MEMBER=FIMS	
	OPER=NO	
		ID=80
	FIRST=ALPHANUM	
IBMOPC	POSIT=60	OTHER=ANY
		MAXLNTH=60
		DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ALPHA	
IDTDATA	POSIT=606	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
		SLBLREQ=NO
		RVRSMAC=NO and EQUALMAC=NO
	OPER=NO	KEYQUAL=0
	PROFDEF=YES	ID=1
	FIRST=NONATNUM	CASE=UPPER
	SIGNAL=NO	
		GENERIC=ALLOWED

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
IDIDMAP	POSIT=591	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	OPER=NO	KEYQUAL=0
		ID=1
	FIRST=ANY	GENERIC=DISALLOWED
IIMS	POSIT=575	OTHER=ALPHANUM
	RACLIST=DISALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	GROUP=JIMS	
	OPER=NO	
		ID=1
	FIRST=ALPHA	
ILMADMIN	POSIT=566	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=ALLOWED	DFTRETC=4
		DFTUACC=NONE
		SLBLREQ=NO
	OPER=NO	
		ID=1
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
INFOMAN	POSIT=85	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=44
	GENLIST=ALLOWED	DFTRETC=4
	GROUP=GINFOMAN	
	OPER=NO	
		ID=103
	FIRST=ANY	
IPCOBJ	POSIT=62	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
	OPER=NO	
	PROFDEF=NO;	ID=1
	FIRST=ANY	
IZP	POSIT=608	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=NO
		RVRSMAC=NO
	OPER=NO	KEYQUAL=0
	PROFDEF=YES	ID=1
	FIRST=NONATABC	CASE=UPPER
	SIGNAL=NO	
		GENERIC=ALLOWED

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
JAVA	POSIT=556	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
		DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ANY	
JCICSJCT	POSIT=5	OTHER=ANY
		MAXLNTH=17
		DFTRETC=4
		DFTUACC=NONE
	GROUP=KCICSJCT	
	OPER=NO	
		ID=29
	FIRST=ANY	
JESINPUT	POSIT=108	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
		EQUALMAC=YES
	OPER=NO	
		ID=67
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
JESJOBS	POSIT=109	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=39
	GENLIST=ALLOWED	DFTRETC=8
		DFTUACC=NONE
	OPER=NO	
		ID=66
	FIRST=ANY	MAXLENX = 246
JESSPOOL	POSIT=110	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=53
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
	OPER=NO	
		ID=65
	FIRST=ANY	
JIMS	POSIT=575	OTHER=ALPHANUM
	RACLIST=DISALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	MEMBER=IIMS	
	OPER=NO	
		ID=1
	FIRST=ALPHA	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
KCICSJCT	POSIT=5	OTHER=ANY
		MAXLNTH=17
		DFTRETC=4
		DFTUACC=NONE
	MEMBER=JCICSJCT	
	OPER=NO	
		ID=30
	FIRST=ANY	
KERBLINK	POSIT=565	OTHER=ANY
		MAXLNTH=240
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	RACLIST=DISALLOWED	
	OPER=NO	
	CASE=ASIS	ID=1
	FIRST=ANY	GENERIC=DISALLOWED
KEYSMSTR	POSIT=543	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=39
	GENLIST=ALLOWED	DFTRETC=8
		DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ALPHANUM	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
LDAP	POSIT=593	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=NO
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	CASE=UPPER
LDAPBIND	POSIT=571	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ANY	
LFSCCLASS	POSIT=12	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
		DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ALPHANUM	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
LIMS	POSIT=576	OTHER=ALPHANUM
	RACLIST=DISALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	GROUP=MIMS	
	OPER=NO	
		ID=1
	FIRST=ALPHANUM	
LOGSTRM	POSIT=61	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=26
	GENLIST=ALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ANY	MAXLENX=44
MCICSPPT	POSIT=5	OTHER=ANY
		MAXLNTH=17
		DFTRETC=4
		DFTUACC=NONE
	GROUP=NCICSPPT	
	OPER=NO	
		ID=35
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
MDSNBP	POSIT=536	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GDSNBP	SLBLREQ=YES
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
MDSNCL	POSIT=538	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GDSNCL	SLBLREQ=YES
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
MDSNDB		MAXLENX=246
	POSIT=528	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GDSNDB	SLBLREQ=YES
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
MDSNGV	POSIT=596	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
	GROUP=GDSNGV	
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
		MAXLENX=246
	SIGNAL=YES	
MDSNJR	POSIT=567	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GDSNJR	SLBLREQ=YES
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
		MAXLENX=246
MDSNPK	POSIT=534	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GDSNPK	SLBLREQ=NO
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	SIGNAL=YES	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
MDSNPN	POSIT=533	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GDSNPN	SLBLREQ=YES
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	SIGNAL=YES	
MDSNSC	POSIT=562	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GDSNSC	SLBLREQ=YES
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	MAXLENX=246	
MDSNSG	POSIT=537	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GDSNSG	SLBLREQ=YES
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	MAXLENX=246	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
MDSNSM	POSIT=535	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GDSNSM	SLBLREQ=YES
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	SIGNAL=YES	
MDSNSP	POSIT=561	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GDSNSP	SLBLREQ=YES
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	SIGNAL=YES	MAXLENX=246
MDSNSQ	POSIT=573	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GDSNSQ	SLBLREQ=NO
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	SIGNAL=YES	MAXLENX=246

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
MDSNTB	POSIT=530	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GDSNTB	SLBLREQ=YES
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	SIGNAL=YES	MAXLENX=246
MDSNTS	POSIT=529	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GDSNTS	SLBLREQ=YES
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
MDSNUF	POSIT=560	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GDSNUF	SLBLREQ=YES
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	SIGNAL=YES	MAXLENX=246

Class	Attributes	
MDSNUF	POSIT=560	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=100
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GDSNUF	SLBLREQ=YES
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
	SIGNAL=YES	MAXLNTH=246
MFADEF	POSIT=600	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	SLBLREQ=NO	
	RVRSMAC=NO	
	OPER=NO	KEYQUAL=0
	PROFDEF=YES	ID=1
	FIRST=NONATNUM	CASE=UPPER
	SIGNAL=NO	
	GENERIC=ALLOWED	
	MGMTCLAS	POSIT=123
RACLIST=ALLOWED		MAXLNTH=8
GENLIST=DISALLOWED		DFTRETC=4
DFTUACC=NONE		
OPER=NO		
ID=49		
FIRST=ALPHA		

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
MIMS	POSIT=576	OTHER=ALPHANUM
	RACLIST=DISALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	MEMBER=LIMS	
	OPER=NO	
		ID=1
	FIRST=ALPHANUM	
MQADMIN	POSIT=80	OTHER=ANY
		MAXLNTH=62
		DFTRETC=8
		DFTUACC=NONE
	GROUP=GMQADMIN	
	OPER=NO	
		ID=120
	FIRST=ANY	
MQCHAN	POSIT=58	OTHER=ANY
		MAXLNTH=53
		DFTRETC=8
		DFTUACC=NONE
	GROUP=GMQCHAN	
	OPER=NO	
		ID=1
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
MQCMDS	POSIT=81	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=22
		DFTRETC=8
		DFTUACC=NONE
	OPER=NO	
		ID=122
	FIRST=ANY	
MQCONN	POSIT=82	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=10
		DFTRETC=8
		DFTUACC=NONE
		EQUALMAC=YES
	OPER=NO	
		ID=123
	FIRST=ANY	
MQNLIST	POSIT=79	OTHER=ANY
		MAXLNTH=53
		DFTRETC=8
		DFTUACC=NONE
	GROUP=GMQNLIST	
	OPER=NO	
		ID=118
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
MQPROC	POSIT=78	OTHER=ANY
		MAXLNTH=53
		DFTRETC=8
		DFTUACC=NONE
	GROUP=GMQPROC	
	OPER=NO	
		ID=116
	FIRST=ANY	
MQQUEUE	POSIT=77	OTHER=ANY
		MAXLNTH=53
		DFTRETC=8
		DFTUACC=NONE
	GROUP=GMQQUEUE	
	OPER=NO	
		ID=2
	FIRST=ANY	
MXADMIN	POSIT=586	OTHER=ANY
		MAXLNTH=62
		DFTRETC=8
		DFTUACC=NONE
	GROUP=GMXADMIN	
	OPER=NO	
		ID=1
	FIRST=ANY	CASE=ASIS

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
MXNLIST	POSIT=585	OTHER=ANY
		MAXLNTH=53
		DFTRETC=8
		DFTUACC=NONE
	GROUP=GMXNLIST	
	OPER=NO	
		ID=1
	FIRST=ANY	CASE=ASIS
MXPROC	POSIT=584	OTHER=ANY
		MAXLNTH=53
		DFTRETC=8
		DFTUACC=NONE
	GROUP=GMXPROC	
	OPER=NO	
		ID=1
	FIRST=ANY	CASE=ASIS
MXQUEUE	POSIT=583	OTHER=ANY
		MAXLNTH=53
		DFTRETC=8
		DFTUACC=NONE
	GROUP=GMXQUEUE	
	OPER=NO	
		ID=1
	FIRST=ANY	CASE=ASIS

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
MXTOPIC	POSIT=587	OTHER=ANY
		MAXLNTH=246
		DFTRETC=8
		DFTUACC=NONE
	GROUP=GMXTOPIC	
	OPER=NO	
		ID=1
	FIRST=ANY	CASE=ASIS
NCICSPPT	POSIT=5	OTHER=ANY
		MAXLNTH=17
		DFTRETC=4
		DFTUACC=NONE
	MEMBER=MCICSPPT	
	OPER=NO	
		ID=36
	FIRST=ANY	
NDSLINK	POSIT=554	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	PROFDEF=YES
	DFTRETC=4	ID=1
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
NETCMDS	POSIT=68	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
		DFTRETC=4
		ID=1
	FIRST=ALPHA	
NETSPAN	POSIT=67	OTHER=ANY
		MAXLNTH=8
		DFTRETC=4
		ID=1
	FIRST=ALPHA	
NODES	POSIT=103	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=24
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	MEMBER=NODMBR	
	OPER=NO	
		ID=72
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
NODMBR	POSIT=103	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	GROUP=NODES	
	OPER=NO	
		ID=43
	FIRST=ANY	
NOTELINK	POSIT=553	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=64
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	PROFDEF=YES
	DFTRETC=4	ID=1
	FIRST=ANY	
NVASAPDT	POSIT=97	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=17
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=83
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
OIMS	POSIT=101	OTHER=ALPHANUM
	RACLIST=DISALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	GROUP=WIMS	
	OPER=NO	
		ID=81
	FIRST=ALPHANUM	
OPERCMDs	POSIT=112	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=39
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	OPER=NO	
		ID=63
	FIRST=ANY	
OPTAUDIT	POSIT=609	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	RVRSMAC=NO	SLBLREQ=NO
	OPER=NO	EQUALMAC=NO
	GENERIC=ALLOWED	SIGNAL=YES
	PROFDEF=YES	ID=1
	FIRST=NONATNUM	CASE=UPPER
	KEYQUAL=0	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
PCICSPSB	POSIT=5	OTHER=ANY
		MAXLNTH=17
		DFTRETC=4
		DFTUACC=NONE
	GROUP=QCICSPSB	
	OPER=NO	
		ID=14
	FIRST=ANY	
PERFGRP	POSIT=125	OTHER=NUMERIC
	RACLIST=ALLOWED	MAXLNTH=3
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=47
	FIRST=NUMERIC	
PIMS	POSIT=101	OTHER=ALPHANUM
	RACLIST=DISALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	GROUP=QIMS	
	OPER=NO	
		ID=75
	FIRST=ALPHANUM	

Table 311. Classes supplied by IBM (continued)

Class	Attributes	
PKISERV	POSIT=597	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	OPER=NO	KEYQUAL=0
		ID=1
	FIRST=ANY	CASE=UPPER
PMBR	POSIT=13	OTHER=ALPHANUM
		MAXLNTH=44
		DFTRETC=4
		DFTUACC=NONE
	GROUP=PROGRAM	
	OPER=NO	
		ID=40
	FIRST=ALPHA	
PRINTSRV	POSIT=570	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=17
	GENLIST=ALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ANY	MAXLENX=64

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
PROCACT	POSIT=75	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
	OPER=NO	
	PROFDEF=NO	ID=111
	FIRST=ANY	
PROCESS	POSIT=74	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
	OPER=NO	
	PROFDEF=NO	ID=110
	FIRST=ANY	
PROGRAM	POSIT=13	OTHER=ALPHANUM
		MAXLNTH=8
		DFTRETC=4
		DFTUACC=NONE
	MEMBER=PMBR	
	OPER=NO	
		ID=41
	FIRST=ALPHA	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
PROPCNTL	POSIT=119	OTHER=ALPHANUM
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	OPER=NO	
		ID=56
	FIRST=ALPHANUM	
PSFMPL	POSIT=113	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=39
	GENLIST=DISALLOWED	DFTRETC=8
	RACLREQ=YES	DFTUACC=NONE
	OPER=YES	
		ID=62
	FIRST=ANY	
PTKTDATA	POSIT=76	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=39
		DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	OPER=NO	
		ID=112
	FIRST=ALPHANUM	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
PTKTVAL	POSIT=76	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=39
		DFTRETC=4
	OPER=NO	
		ID=1
	FIRST=ANY	
QCICSPSB	POSIT=5	OTHER=ANY
		MAXLNTH=17
		DFTRETC=4
		DFTUACC=NONE
	MEMBER=PCICSPSB	
	OPER=NO	
		ID=15
	FIRST=ANY	
QIMS	POSIT=101	OTHER=ALPHANUM
	RACLIST=DISALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	MEMBER=PIMS	
	OPER=NO	
		ID=76
	FIRST=ALPHANUM	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
RACFEVNT	POSIT=574	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=ALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ANY	
RACFHC	POSIT=590	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=8
	RACLREQ=YES	DFTUACC=NONE
	MEMBER=RACHCMBR	
	OPER=NO	
		ID=1
RACFVARS	POSIT=102	OTHER=ALPHANUM
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	MEMBER=RVARSMBR	
	OPER=NO	
		ID=74
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
RACGLIST	POSIT=10	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=14
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=NO
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	
RACHCMBR	POSIT=590	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
	GROUP=RACFHC	
	OPER=NO	
		ID=1
	FIRST=ALPHANUM	
RAUDITX	POSIT=577	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
		DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		ID=1
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
RCICSRES	POSIT=5	OTHER=ANY
		MAXLNTH=54
		DFTUACC=NONE
	GROUP=WCICSRES	
	OPER=NO	
		ID=52
	FIRST=ANY	CASE=ASIS
RDATA LIB	POSIT=581	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	OPER=NO	KEYQUAL=0
		ID=1
	FIRST=ANY	CASE=UPPER
REALM	POSIT=564	OTHER=ANY
		MAXLNTH=240
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	RACLIST=ALLOWED	
		DFTRETC=4
	OPER=NO	
		ID=1
	FIRST=ANY	GENERIC=DISALLOWED

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
RIMS	POSIT=589	OTHER=ALPHANUM
		MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	RACLIST=DISALLOWED	
	OPER=NO	ID=1
	FIRST=ALPHANUM	CASE=ASIS
RMTOPS	POSIT=86	OTHER=ANY
		MAXLNTH=246
		DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=102
	FIRST=ALPHA	
RODMMGR	POSIT=69	OTHER=ANY
		MAXLNTH=44
		DFTRETC=4
		ID=113
	FIRST=ALPHANUM	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
ROLE	POSIT=551	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=8
	PROFDEF=YES	DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ANY	
RRSFDATA	POSIT=65	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=ALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	OPER=NO	KEYQUAL=0
		ID=1
	FIRST=ANY	
RVARSMBR	POSIT=102	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=39
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	GROUP=RACFVARS	
	OPER=NO	
		ID=73
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
SCDMBR	POSIT=9	OTHER=ANY
		MAXLNTH=39
		DFTRETC=4
		DFTUACC=NONE
	GROUP=SECDATA	
	OPER=NO	
		ID=25
	FIRST=ANY	
SCICSTST	POSIT=5	OTHER=ANY
		MAXLNTH=17
		DFTRETC=4
		DFTUACC=NONE
	GROUP=UCICSTST	
	OPER=NO	
		ID=33
	FIRST=ANY	MAXLENX=25
SDSF	POSIT=100	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=63
	GENLIST=ALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	GROUP=GSDFS	
	OPER=NO	
		ID=94
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
SECDATA	POSIT=9	OTHER=ALPHA
		MAXLNTH=8
		DFTRETC=4
		DFTUACC=NONE
	MEMBER=SCDMBR	
	OPER=NO	
		ID=26
	FIRST=ALPHA	
SECLABEL	POSIT=117	OTHER=ALPHANUM
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=8
	RACLREQ=YES	DFTUACC=NONE
	MEMBER=SECLMBR	
	OPER=NO	
		ID=58
	FIRST=ALPHA	
SECLMBR	POSIT=117	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=4
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	GROUP=SECLABEL	
	OPER=NO	
		ID=1
	FIRST=ANY	GENERIC=DISALLOWED

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
SERVAUTH	POSIT=558	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=64
		DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
		SLBLREQ=YES
		EQUALMAC=YES
	OPER=NO	
		ID=1
	FIRST=ALPHA	
	SIGNAL=YES	
SERVER	POSIT=546	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=41
		DFTRETC=8
		SLBLREQ=YES
		EQUALMAC=YES
	OPER=NO	
		ID=1
	FIRST=ALPHANUM	MAXLENX=64
SFSCMD	POSIT=99	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
	OPER=NO	
		ID=93
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
SIMS	POSIT=101	OTHER=ALPHANUM
	RACLIST=DISALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	GROUP=UIMS	
	OPER=NO	
		ID=77
	FIRST=ALPHANUM	
SMESSAGE	POSIT=116	OTHER=ALPHANUM
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=0
		DFTUACC=NONE
	OPER=NO	
		ID=59
	FIRST=ALPHANUM	
SOMDOBJs	POSIT=547	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
		DFTRETC=8
	GROUP=GSOMDOBJ	
	OPER=NO	
		ID=1
	FIRST=ALPHANUM	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
STARTED	POSIT=66	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=17
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ALPHA	
STORCLAS	POSIT=122	OTHER=ALPHANUM
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=50
	FIRST=ALPHA	
SUBSYSNM	POSIT=83	OTHER=ALPHANUM
	RACLIST=ALLOWED	MAXLNTH=8
		DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ALPHA	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
SURROGAT	POSIT=104	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=17
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=71
	FIRST=ANY	
SYSMVIEW	POSIT=542	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=52
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ALPHANUM	
SYSAUTO	POSIT=598	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
		SLBLREQ=NO
	OPER=NO	KEYQUAL=0
	PROFDEF=YES	ID=1
	FIRST=ALPHANUM	CASE=UPPER
	SIGNAL=NO	GENERIC=ALLOWED

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
TAPEVOL	POSIT=1	OTHER=ALPHANUM
		MAXLNTH=6
		DFTRETC=4
		SLBLREQ=YES
	OPER=YES	
		ID=6
	FIRST=ALPHANUM	
TCICSTRN	POSIT=5	OTHER=ANY
		MAXLNTH=13
		DFTRETC=4
		DFTUACC=NONE
	GROUP=GCICSTRN	
	OPER=NO	
		ID=12
	FIRST=ANY	
TEMPDSN	POSIT=106	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=39
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
	OPER=NO	
	PROFDEF=NO	ID=69
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
TERMINAL	POSIT=2	OTHER=ALPHANUM
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=ALLOWED	DFTRETC=4
	GROUP=GTERMINL	SLBLREQ=YES
		EQUALMAC=YES
	OPER=NO	
		ID=7
	FIRST=ALPHANUM	
	SIGNAL=YES	
TIMS	POSIT=4	OTHER=ALPHANUM
		MAXLNTH=8
		DFTRETC=4
		DFTUACC=NONE
	GROUP=GIMS	
	OPER=NO	
		ID=9
	FIRST=ALPHANUM	
TMEADMIN	POSIT=549	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=ALLOWED	DFTRETC=8
		DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ALPHANUM	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
TSOAUTH	POSIT=124	OTHER=ALPHANUM
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=48
	FIRST=ALPHANUM	
TSOPROC	POSIT=127	OTHER=ALPHANUM
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=45
	FIRST=ALPHA	
UCICSTST	POSIT=5	OTHER=ANY
		MAXLNTH=17
		DFTRETC=4
		DFTUACC=NONE
	MEMBER=SCICSTST	
	OPER=NO	
		ID=34
	FIRST=ANY	MAXLENX=25

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
UIMS	POSIT=101	OTHER=ALPHANUM
	RACLIST=DISALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	MEMBER=SIMS	
	OPER=NO	
		ID=78
	FIRST=ALPHANUM	
UNIXMAP	POSIT=552	OTHER=NUMERIC
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=1
	FIRST=ALPHA	
UNIXPRIV	POSIT=555	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	RACLREQ=YES	SLBLREQ=NO
	OPER=NO	
		ID=1
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
VCICSCMD	POSIT=5	OTHER=ANY
		MAXLNTH=21
		DFTRETC=4
		DFTUACC=NONE
	MEMBER=CCICSCMD	
	OPER=NO	
		ID=53
	FIRST=ANY	
VMBATCH	POSIT=15	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=ALLOWED	DFTRETC=4
		DFTUACC=NONE
		ID=24
	FIRST=ANY	
VMBR	POSIT=120	OTHER=ALPHANUM
	RACLIST=DISALLOWED	MAXLNTH=39
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	GROUP=VMEVENT	
	OPER=NO	
		ID=54
	FIRST=ALPHA	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
VMCMD	POSIT=14	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=17
	GENLIST=ALLOWED	DFTRETC=4
		DFTUACC=NONE
		ID=22
	FIRST=ANY	
VMDEV	POSIT=594	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=ALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=YES
		RVRSMAC=NO
	OPER=NO	KEYQUAL=0
	PROFDEF=YES	ID=1
	FIRST=ANY	
VMEVENT	POSIT=120	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=16
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	MEMBER=VMBR	
	OPER=NO	
		ID=55
	FIRST=ALPHA	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
VMLAN	POSIT=64	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=ALLOWED	DFTRETC=4
		DFTUACC=NONE
		SLBLREQ=YES
	OPER=NO	
		ID=1
	FIRST=ANY	
VMMAC	POSIT=91	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
		SLBLREQ=YES
	OPER=NO	
	PROFDEF=NO	ID=91
	FIRST=ANY	
VMMDISK	POSIT=18	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=22
	GENLIST=ALLOWED	DFTRETC=4
		DFTUACC=NONE
		SLBLREQ=YES
		ID=20
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
VMNODE	POSIT=16	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=ALLOWED	DFTRETC=4
		DFTUACC=NONE
		ID=23
	FIRST=ANY	
VMPOSIX	POSIT=63	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	OPER=NO	
		ID=1
VMRDR	POSIT=17	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=17
	GENLIST=ALLOWED	DFTRETC=4
		DFTUACC=NONE
		ID=21
	FIRST=ANY	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
VMSEGMT	POSIT=90	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=ALLOWED	DFTRETC=4
		DFTUACC=NONE
		SLBLREQ=YES
	OPER=NO	
		ID=92
	FIRST=ANY	
VMXEVENT	POSIT=96	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=16
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	MEMBER=VXMBR	
	OPER=NO	
		ID=85
	FIRST=ALPHA	
VTAMAPPL	POSIT=114	OTHER=ALPHANUM
	RACLIST=ALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
	OPER=NO	
		ID=61
	FIRST=ALPHANUM	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
VXMBR	POSIT=96	OTHER=ALPHANUM
	RACLIST=DISALLOWED	MAXLNTH=39
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	GROUP=VMXEVENT	
	OPER=NO	
		ID=84
	FIRST=ALPHA	
WBEM	POSIT=604	OTHER=NONATNUM
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=8
	RACLREQ=YES	DFTUACC=NONE
		SLBLREQ=NO
		RVRSMAC=NO
	OPER=NO	KEYQUAL=0
	PROFDEF=YES	ID=1
	FIRST=NONATABC	CASE=UPPER
	SIGNAL=NO	
		GENERIC=ALLOWED
WCICSRES	POSIT=5	OTHER=ANY
		MAXLNTH=40
		DFTUACC=NONE
	MEMBER=RCICSRES	
	OPER=NO	
		ID=53
	FIRST=ANY	CASE=ASIS

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
WIMS	POSIT=101	OTHER=ALPHANUM
	RACLIST=DISALLOWED	MAXLNTH=8
	GENLIST=DISALLOWED	DFTRETC=4
		DFTUACC=NONE
	MEMBER=OIMS	
	OPER=NO	
		ID=82
	FIRST=ALPHANUM	
WRITER	POSIT=111	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=39
	GENLIST=DISALLOWED	DFTRETC=8
		DFTUACC=NONE
		SLBLREQ=YES
		RVRSMAC=YES
	OPER=NO	
		ID=64
	FIRST=ANY	
XCSFKEY	POSIT=98	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=8
	RACLREQ=YES	DFTUACC=NONE
	GROUP=GXCSEKEY	
	OPER=NO	
		ID=100
	FIRST=ALPHA	
	SIGNAL=YES	

Table 311. Classes supplied by IBM (continued)		
Class	Attributes	
XFACILIT	POSIT=8	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=ALLOWED	DFTRETC=8
		DFTUACC=NONE
	GROUP=GXFACILI	
	OPER=NO	
		ID=1
	FIRST=ALPHANUM	
	SIGNAL=YES	
ZMFAPLA	POSIT=592	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
	GROUP=GZMFAPLA	SLBLREQ=NO
	OPER=NO	
	PROFDEF=YES	ID=1
	FIRST=ANY	CASE=ASIS
ZMFCLOUD	POSIT=602	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=8
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=NO
		RVRSMAC=NO, EQUALMAC=NO
	OPER=NO	KEYQUAL=0
	PROFDEF=YES	ID=1
	FIRST=ANY	CASE=ASIS
	SIGNAL=YES	
		GENERIC=ALLOWED

<i>Table 311. Classes supplied by IBM (continued)</i>		
Class	Attributes	
ZOWE	POSIT=607	OTHER=ANY
	RACLIST=DISALLOWED	MAXLNTH=246
	GENLIST=DISALLOWED	DFTRETC=4
	RACLREQ=NO	DFTUACC=NONE
		SLBLREQ=NO
		RVRSMAC=NO, EQUALMAC=NO
	OPER=NO	KEYQUAL=0
	PROFDEF=YES	ID=1
	FIRST=NONATABC	CASE=UPPER
	SIGNAL=NO	
		GENERIC=ALLOWED

Appendix D. RACF database templates

This topic includes information about the following templates of the RACF database:

- GROUP
- USER
- CONNECT
- DATA SET
- GENERAL
- RESERVED

Important

Do not modify the RACF database templates (CSECT IRRTEMP2). Such modification is not supported by IBM and might result in damage to your RACF database or other unpredictable results.

Segment fields:

1. The first field in a segment of a template cannot be retrieved or updated. This field has a Field ID of 001 and is usually described in the **Field Being Described** column as 'Start of segment fields'.
2. The TME segment fields are intended to be updated by Tivoli® applications, which manage updates, permissions, and cross-references among the fields. The TME fields should only be directly updated on an exception basis. See *z/OS Security Server RACF Command Language Reference* for formats of the field data as enforced by the RACF commands. Use caution when directly updating TME fields, as the updates might be overridden by subsequent actions of Tivoli applications.

Format of field definitions

The RACF database templates contain a definition for each field in the profile.

Each field definition contains information about the field in the following format:

Field name (character data)	Field ID	Flag 1	Flag 2	Field Length decimal	Default value	Type
Field Name		Character data.				
Field ID		Reference number.				
Flag 1 field		The bits have the following meanings when they are turned on:				
	Bit 0:	The field is a member of a repeat group.				
	Bit 1:	The definition describes a combination field.				
	Bit 2:	The field is a flag byte.				
	Bit 3:	The field contains the count of members in the repeat group following this field.				
	Bit 4:	The definition describes a combination field continued in next entry.				
	Bit 5:	The field (for example, PASSWORD) is encrypted.				
	Bit 6:	The field is sorted in ascending order.				
	Bit 7:	The field is a statistical field. A value is always stored for this field, even when it is equal to the defined null value for the field.				
Flag 2		The bits have the following meanings when they are turned on:				
	Bit 0:	Changes to this field affect security and cause ACEEs to be purged from VLF.				

Field name (character data)	Field ID	Flag 1	Flag 2	Field Length decimal	Default value	Type
	Bit 1:	The field is padded on the left with binary zeros when values shorter than the field length are retrieved.				
	Bit 2:	This field represents a 3-byte date field.				
	Bit 3:	This field is an Application Identity Mapping alias name.				
	Bit 4:	This field is not to be unloaded by the Database Unload utility (IRRDBU00).				
	Bit 5:	The alias name in this field is EBCDIC.				
	Bits 6–7:	Reserved for IBM's use.				
Field Length		Field length on return from ICHEINTY or RACROUTE REQUEST=EXTRACT (0 is variable length).				
Default Value		Field default. If the field is not present in the profile, this byte is propagated throughout the returned field as the default value.				
Type		<p>Data type of each field. In this column, character is represented as 'Char', integer is represented as 'Int', and binary is represented as 'Bin'. 'Date' and 'Time' are also possible data types.</p> <p>The type of a combination field that represents a single field is the same as that single field. There is no "type" associated with a combination field which represents multiple fields.</p>				

Repeat groups on the RACF database

A repeat group consists of one or more sequential fields within a profile that are able to be repeated within that profile. A field that belongs to a repeat group is only defined once in the template, but can be repeated as many times as necessary within the actual profile. A count field precedes the repeat group in the profile indicating how many of these groups follow.

Field length

If a field in a profile has a fixed length, a value (less than 255) in the field definition within the template specifies its actual length. If a field in a profile has a variable length, the value in the field definition is 0. In both cases, the actual field length is contained in the physical data mapped by the field definition.

Data field types

RACF stores information in the RACF database in many different formats. This section identifies the major data types that RACF stores. Exceptions and additional detail can be found in the description of each specific field within the templates.

Date fields

The format of the 3-byte date fields is *yydddF*, which represents a packed decimal number in which *y* represents year, *d* represents day, and *F* represents the sign. Examples of RACF date values are X'98111C' and X'94099D'.

The format of the 4-byte date fields should be *yyyymmdd*, which represents a packed decimal number in which *y* represents year, *m* represents month, and *d* represents day. Examples of RACF date values are X'19980421' and X'19940409'.

RACF might use any of the following values for null dates: X'FFFFFF', X'00000D', X'00000C', and X'000000' for 3-byte addresses, and X'FFFFFFFF', X'0000000D', X'0000000C', and X'00000000' for 4-byte addresses. However, you should always set null dates to either X'00000F' for 3-byte addresses and X'0000000F' for 4-byte addresses.

Time fields

The format for the 4-byte time fields are *hhmmssstc* where *h* represents hours, *m* represents minutes, *s* represents seconds, *t* represents tenths of seconds, and *c* represents hundredths of seconds. There is no sign byte. For information on the 8-byte version, see the TIME macro as documented in *z/OS MVS Programming: Assembler Services Reference IAR-XCT*.

Integer fields

Integers are stored as unsigned binary values. These values can be 1, 2, or 4 bytes in length.

Character fields

Character fields are padded with blanks to the right.

Combination fields on the RACF database

The database templates also contain definitions called *combination fields*.

Combination fields do not describe a field of a profile. They contain the field numbers that identify the respective field definitions. You can use a combination as an alias to access multiple fields with one ICHEACTN or RACROUTE REQUEST=EXTRACT macro. For more information, see "Example 2: Adding a user ID to a data set access list", in Appendix A of *z/OS Security Server RACF Macros and Interfaces*.

In addition, you can use the combination field to provide aliases for individual fields.

The format of a combination field definition is different from a non-combination definition. Its format is as follows:

Field Attribute	Description
Field Name	Character data.
Field ID	Reference number.
Flag 1	The hex representation of the flag bits for this field. For combination fields, bit 1 is on. For a continuation of combination fields, bit 4 is also on.
Flag 2	The hex representation of the flag bits for this field. For combination fields, all bits are off.
Combination IDs	If nonzero, combination IDs represent the position of a non-combination field within the template segment. There can be up to 5 field IDs representing the template fields that comprise this combination field.
Type	Data type of each field. In this column, character is represented as 'Char', integer is represented as 'Int', and binary is represented as 'Bin'. 'Date' and 'Time' are also possible data types.
Comments	Comment field.

Determining space requirements for the profiles

The formula for calculating the space that is required for each segment (Base RACF information, TSO, DFP, and so on) of each profile in the RACF database is as follows:

$$P = 20 + L + F1 + F4 + R$$

Where:

		Space required
P	=	The number of bytes required for a profile segment
L	=	The number of bytes in the profile name

		Space required
F1	=	<p>The sum of the lengths of all fields that contain data and have a length of 1 to 127 bytes, plus 2 bytes for every field counted.</p> <p>For example, if a segment contains 3 non-null fields of length 8, $F1 = (3 * 8) + (3 * 2) = 24 + 6 = 30$.</p>
F4	=	<p>The sum of the lengths of all fields that contain data and have a length of 128 to 2^{31} bytes, plus 5 bytes for every field counted.</p> <p>For example, if a segment contains a non-null field 150 bytes long and a non-null field 255 bytes long, $F4 = 150 + 255 + (2 * 5) = 150 + 255 + 10 = 415$</p>
R	=	<p>The sum of the lengths of all repeat groups. If a repeat group has no occurrences, then it has a length of 0 bytes. If a repeat group has 1 or more occurrences, then the length of each repeat group is calculated as follows:</p> $9 + N + G1 + G4$ <p>N = The number of occurrences of the group.</p> <p>G1 = The sum of the lengths of all fields in the group, which have a length of 1 to 127 bytes, plus 1 byte for every field counted. If a field has a length of zero, it still takes up 1 byte in the profile.</p> <p>G4 = The sum of the lengths of all fields in the group, which have a length of 128 to 2^{31} bytes, plus 4 bytes for every field counted.</p> <p>For example, consider a group with two occurrences. Each occurrence contains an 8-byte field and a variable length field. In the first occurrence, the variable length field is 30 bytes and in second occurrence, it is 200 bytes. The length of the group is: $9 + 2 + G1 + G4$</p> <p>G1 is $(8 + 1) + (30 + 1)$ from the first occurrence and $(8 + 1)$ from the second, for a total of 49 bytes. G4 is $(200 + 4)$ from the second occurrence, or 204 bytes. So, the length of the group is $9 + 2 + 49 + 204$, or 264 bytes.</p>

Note: For each repeat group (except CGGRPCT in the USER profile), the amount of data cannot exceed 65 535 bytes to ensure proper processing by programs retrieving the data using ICHEINTY with DATAMAP=OLD. To calculate the amount of data to determine whether it fits within this limit, examine the template definitions for the repeat group and the data for that repeat group contained within the profile. For each fixed length field in each occurrence of the repeat group, add the length of the field as shown in its template definition. For each variable length field in each occurrence of the repeat group, add the length of the data in the field plus one. When you are done, the total cannot exceed 65 535.

For example, this would translate into a maximum of 8191 group connections per user, based on the CONGRPCT repeat group in the USER template. This group contains one 8-byte field, making the calculation of the limit a simple one of dividing 65 535 by 8 and dropping any remainder.

As another example, this would translate into a maximum of 5957 users connected to a group, based on the ACLCNT repeat group in the GROUP template. This group contains one 8-byte field (USERID), one 1-byte field (USERACS), and one 2-byte field (ACSCNT). This gives a total length of eleven for the fixed-length fields in each occurrence. Dividing 65 535 by 11 and dropping the remainder gives the limit of 5957.

When calculating F1 and F4, remember that statistical fields (Flag1/bit 7 on, in the template definition) are always stored in a profile segment, even when the field contains a null value. For example, LJTIME always adds 3 bytes to the length of a USER profile Base segment, regardless of whether it contains a zero value or some other value. Other fields only exist in the segment if a specific value has been added for that field.

Note: The RACF database space required for a segment is a multiple of the 256-byte slots required to contain the segment. For example, if a USER profile Base segment contains 188 bytes of data, it still requires 256 bytes of space in the RACF database.

Determining space requirements for alias index entries

An exact formula for calculating the space required for alias entries cannot be derived due to index block compression, and the mechanics of the higher-level index blocks. The following is an approximate formula for space taken up in the alias index block sequence set (level 1) by an index entry:

$$AIE = 16 + (N * (2 + BPL)) + AKL$$

Where:

		Space taken up
AIE	=	Alias index entry length
N	=	The number of base profile names. Most aliases are only allowed 1 base profile association.
BPL	=	Base profile name length. The base profile is named in the alias index entry. Because the base profile is a user or group profile, valid BPL values are between 1 and 8.
AKL	=	Alias index entry, key length. The first 3 bytes are template number, segment number, and field number. Additional bytes of the alias index key are taken up by the alias value. These lengths vary according to each alias field.

Group template for the RACF database

The group template describes the fields of group profiles in the RACF database.

NOT Programming Interface Information			
ACSCNT FIELD	FLDCNT FLDFLAG	FLDNAME FLDVALUE	INITCNT
End NOT Programming Interface Information			

Note:

1. Application developers should not depend on being able to use RACROUTE REQUEST=EXTRACT for the BASE segment fields on any security product other than RACF. These products are expected to support only such segments as DFP and TSO.
2. The TME segment fields are intended to be updated by the Tivoli applications, which manage updates, permissions, and cross-references among the fields. The TME fields should only be directly updated on an exception basis. See *z/OS Security Server RACF Command Language Reference* for formats of the field data as enforced by the RACF commands. Use caution when directly updating TME fields, as the updates might be overridden by subsequent actions of Tivoli applications.

The contents of the group template are as follows:

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
The following is the BASE segment of the GROUP template.							
GROUP	001	00	00	00000000	00		
ENTYPE	002	00	00	00000001	01	Int	The number (1) corresponding to group profiles.
VERSION	003	00	00	00000001	01	Int	The version field from the profile. Always X'01'.
SUPGROUP	004	00	80	00000008	FF	Char	The superior group to this group.

Group template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
AUTHDATE	005	00	20	00000003	FF	Date	The date the group was created.
AUTHOR	006	00	80	00000008	FF	Char	The owner (user ID or group name) of the group.
INITCNT	007	00	00	00000002	FF		Reserved for IBM's use.
UACC	008	20	00	00000001	00	Bin	<p>The universal group authority. (The authority of a user to the group if the user is not connected to the group.)</p> <p>Bit</p> <p>Meaning when set</p> <p>0 JOIN authority</p> <p>1 CONNECT authority</p> <p>2 CREATE authority</p> <p>3 USE authority</p> <p>4–7 Reserved for IBM's use</p> <p>Note: This field has a value of X'00', except for the IBM-defined group VSAMDSET, where the value is X'20'.</p>
NOTRMUAC	009	20	00	00000001	00	Bin	If bit 0 is on, the user must be specifically authorized (by the PERMIT command) to use the terminal. If off, RACF uses the terminal's UACC.
INSTDATA	010	00	00	00000000	00	Char	Installation data.
MODELNAM	011	00	00	00000000	00	Char	Data set model profile name. The profile name begins with the second qualifier; the high-level qualifier is not stored.
FLDCNT	012	10	00	00000004	00		Reserved for IBM's use.
FLDNAME	013	80	00	00000008	00		Reserved for IBM's use.
FLDVALUE	014	80	00	00000000	00		Reserved for IBM's use.
FLDFLAG	015	A0	00	00000001	00		Reserved for IBM's use.
SUBGRPCT	016	10	00	00000004	00	Int	The number of subgroups of the group.
SUBGRPNM	017	80	80	00000008	00	Char	A list of the subgroup names.
ACLCNT	018	10	00	00000004	00	Int	The number of users connected to the group.
USERID	019	80	00	00000008	00	Char	The user ID of each user connected to the group.

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
USERACS	020	A0	00	00000001	00	Bin	The group authority of each user connected to the group. Bit Meaning when set 0 JOIN authority 1 CONNECT authority 2 CREATE authority 3 USE authority 4–7 Reserved for IBM's use
ACSCNT	021	80	00	00000002	00		Reserved for IBM's use.
USRCNT	022	10	00	00000004	00	Int	Reserved for installation use. See Note 1 .
USRNM	023	80	00	00000008	00		Reserved for installation use. See Note 1 .
USRDATA	024	80	00	00000000	00		Reserved for installation use. See Note 1 .
USRFLG	025	A0	00	00000001	00		Reserved for installation use. See Note 1 .
UNVFLG	026	20	00	00000001	00	Bin	Identifies the group as having (bit 0 is on) or not having the UNIVERSAL attribute.

Note 1: Intended usage for these fields is to allow the installation to store additional data in this profile. USRNM should have a field name to use as a key to identify each unique occurrence of a row in the repeat group. USRDATA and USRFLG hold the data associated with that name. For more information, see *Example 5: Updating the installation fields*, in *Examples of ICHEINTY, ICHETEST, and ICHEACTN macro usage in z/OS Security Server RACF Macros and Interfaces*.

Field name	Field ID	Flag 1	Flag 2	Combination field IDs					Type	
The following are the COMBINATION fields of the GROUP template.										
DEFDATE	000	40	00	005	000	000	000	000	Char	Alias for AUTHDATE
CREADATE	000	40	00	005	000	000	000	000	Char	Alias for AUTHDATE
OWNER	000	40	00	006	000	000	000	000	Char	Alias for AUTHOR
FIELD	000	40	00	013	014	015	000	000		FLDNAME, FLDVALUE, and FLDFLAG
ACL	000	40	00	019	020	021	000	000		USERID, USERACS, and ACSCNT
USERDATA	000	40	00	023	024	025	000	000		USERNM, USERDATA, and USERFLG

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
The following is the DFP segment of the GROUP template.							
DFP	001	00	00	00000000	00		Start of segment
DATAAPPL	002	00	00	00000000	00	Char	Data Application
DATACLAS	003	00	00	00000000	00	Char	Data Class
MGMTCLAS	004	00	00	00000000	00	Char	Management Class

User template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
STORCLAS	005	00	00	00000000	00	Char	Storage Class
The following is the OMVS segment of the GROUP template.							
OMVS	001	00	00	00000000	00		Start of segment
GID	002	00	10	00000004	FF	Int	GID
The following is the OVM segment of the GROUP template.							
OVM	001	00	00	00000000	00		Start of segment
GID	002	00	00	00000004	FF	Int	GID
The following is the TME segment of the GROUP template.							
TME	001	00	00	00000000	00		Start of segment fields
ROLEN	002	10	00	00000004	00	Int	Count of roles
ROLES	003	80	00	00000000	00	Char	Role names
The following is the CSDATA segment of the GROUP template.							
CSDATA	001	00	00	0	0		Start of segment fields for custom fields Note: Intended usage for these fields is dictated by your installation. See <i>z/OS Security Server RACF Security Administrator's Guide</i> for more information on custom fields.
CSCNT	002	10	00	4	00	Integer	Count of custom fields
CSTYPE	003	80	00	1	01	Bin	Custom field type: <ul style="list-style-type: none">• 01 - character• 02 - numeric• 03 - flag• 04 - hex
CSKEY	004	80	00	00	00	Char	Custom field keyword; maximum length = 8
CSVALUE	005	80	00	0	00	Char	Custom field value

Field name	Field ID	Flag 1	Flag 2	Combination field IDs						Type	
The following is a COMBINATION field of the CSDATA segment of the GROUP template.											
CSCDATA	000	40	00	003	004	005	000	000	Char		Combination field for custom fields

User template for the RACF database

The user template describes the fields of the user profiles in a RACF database.

NOT Programming Interface Information			
CATEGORY CONGRPCT CONGRPNM CURKEY CURKEYV ENCTYPE FACACDT FACTAGS FACTOR FACTORN FIELD FLDCNT	FLDFLAG FLDNAME FLDVALUE MAGSTRIP MFAFLBK MFAPOLN MFAPOLNM NUMCTGY OLDPHR OLDPHRES OLDPHREX OLDPHRNM OLDPHRNX OLDPHRX	OLDPWD OLDPWDNM OLDPWDX OPWDX OPWDXCT OPWDXGEN PASSWORD PHRASE PHRASEX PHRCNT PHRCNTX PHRGEN	PPHENV PREVKEY PREVKEYV PWDCNT PWDENV PWDGEN PWDX SALT
End NOT Programming Interface Information			

Notes:

1. Application developers should not depend on being able to use RACROUTE REQUEST=EXTRACT for the BASE segment fields on any security product other than RACF. These [“Reserved template for the RACF database”](#) on page 652e products are expected to support only such segments as DFP and TSO.
2. PASSWORD and PHRASE are not programming interface fields when KDFAES is the active encryption algorithm.

The contents of the user template (base segment) are as follows:

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
The following is the BASE segment of the USER template.							
USER	001	00	00	00000000	00		
ENTYPE	002	00	00	00000001	02	Int	The number (2) corresponding to user profiles.
VERSION	003	00	00	00000001	01	Int	The version field from the profile. Always X'01'.
AUTHDATE	004	00	20	00000003	FF	Date	The date the user was defined to RACF.
AUTHOR	005	00	00	00000008	FF	Char	The owner (user ID or group name) of the user profile.
FLAG1	006	20	80	00000001	00	Bin	Identifies the user as having (bit 0 is on) or not having the ADSP attribute.
FLAG2	007	20	80	00000001	00	Bin	Identifies the user as having (bit 0 is on) or not having the SPECIAL attribute.
FLAG3	008	20	80	00000001	00	Bin	Identifies the user as having (bit 0 is on) or not having the OPERATIONS attribute.
FLAG4	009	20	80	00000001	00	Bin	Identifies the user as having (bit 0 is on) or not having the REVOKE attribute.
FLAG5	010	20	80	00000001	00	Bin	Identifies the user as having (bit 0 is on) or not having the GRPACC attribute.

User template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
PASSINT	011	00	80	00000001	FF	Int	The interval in days (represented by a number between 1 and 254) that the user's password is in effect. If it is X'FF', the user's password never expires. See the description of the SETR PASSWORD(INTERVAL...)) processing instructions in z/OS Security Server RACF Command Language Reference for more details.
PASSWORD	012	04	80	00000008	FF	Char	The password associated with the user. For masking, the masked password is stored. For DES or KDFAES, the encrypted user ID is stored. If the installation provides its own password authentication, data returned by the ICHDEX01 exit is stored.
PASSDATE	013	00	A0	00000003	FF	Date	The date of password change.
PGMRNAME	014	00	00	00000020	FF	Char	The name of the user.
DFLTGRP	015	00	00	00000008	FF	Char	The default group associated with the user. A value of X'FF' indicates that no group was specified.
LJTIME	016	01	00	00000004	FF	Time	The last recorded time that the user entered the system by using RACROUTE REQUEST=VERIFY.
LJDATE	017	01	20	00000003	FF	Date	The last recorded date that the user entered the system by using RACROUTE REQUEST=VERIFY.
INSTDATA	018	00	80	00000000	00	Char	Installation data.
UAUDIT	019	20	80	00000001	00	Bin	Identifies whether all RACROUTE REQUEST=AUTH, RACROUTE REQUEST=DEFINE, (and, if the caller requests logging, RACROUTE REQUEST=FASTAUTH) macros issued for the user and all RACF commands (except SEARCH, LISTDSD, LISTGRP, LISTUSER, and RLIST) issued by the user is logged. If bit 0 is on, they are logged. If bit 0 is off, logging might still occur for other reasons, as identified in z/OS Security Server RACF Auditor's Guide .
FLAG6	020	20	80	00000001	00	Bin	Identifies the user as having (bit 0 is on) or not having the AUDITOR attribute.
FLAG7	021	20	80	00000001	00	Bin	If bit 0 is on, and FLAG8 has bit 0 on, an operator identification card (OID card) is needed to enter the system. If bit 1 is on, this is a protected user ID, which cannot enter the system by any means requiring a password or OID card. If bit 2 is on, this user can enter the system with a password phrase.
FLAG8	022	20	80	00000001	00	Bin	If bit 0 is on, an operator identification card (OID card) is required when logging on to the system.
MAGSTRIP	023	04	00	00000000	00	Bin	The operator identification associated with the user from the masked or encrypted OID card data required to authenticate this user, as supplied by a supported 327x (such as 3270 and 3278) OID card reader.
PWDGEN	024	00	00	00000001	FF	Int	Current password generation number.
PWDCNT	025	10	00	00000004	00	Int	Number of old passwords present.
OLDPWDNM	026	80	00	00000001	00	Int	Generation number of previous password.

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
OLDPWD	027	84	00	00000008	FF	Char	Previous password. This is an encrypted password value.
REVOECT	028	01	80	00000001	FF	Int	Count of unsuccessful password attempts. Note: You can use ALTER when setting this field, but you cannot use ALTERI.
MODELNAM	029	00	80	00000000	00	Char	Data set model profile name. The profile name begins with the second qualifier; the high-level qualifier is not stored.
SECLEVEL	030	00	80	00000001	FF	Int	The number that corresponds to the user's security level. For more information on security levels, see z/OS Security Server RACF Security Administrator's Guide .
NUMCTGY	031	10	80	00000004	00	Int	Number of security categories.
CATEGORY	032	80	80	00000002	00	Int	A number that corresponds to the security categories to which the user has access.
REVOKEDT	033	00	20	00000000	00	Date	The date the user is revoked. This field either has length 0, or contains a 3-byte revoke date.
RESUMEDT	034	00	20	00000000	00	Date	The date the user is resumed. This field either has length 0, or contains a 3-byte resume date.
LOGDAYS	035	20	00	00000001	00	Bin	The days of the week the user cannot log on (Bit 0 of this field equals Sunday, bit 1 equals Monday, and so on).
LOGTIME	036	00	80	00000000	00	Time	The time of the day the user can log on. If present (length of variable field not equal to 0), it is specified as 6 bytes formatted as two 3-byte packed decimal fields, 0ssssC0eeeeC, where ssss represents the start time (hhmm) from the ALU...WHEN(TIMES(...)) specification and eeee represents the end time. For hhmm, hh represents hours, and mm represents minutes.
FLDCNT	037	10	00	00000004	00		Reserved for IBM's use.
FLDNAME	038	80	00	00000008	00		Reserved for IBM's use.
FLDVALUE	039	80	00	00000000	00		Reserved for IBM's use.
FLDFLAG	040	A0	00	00000001	00		Reserved for IBM's use.
CLCNT	041	10	80	00000004	00	Int	The number of classes in which the user is allowed to define profiles.
CLNAME	042	80	80	00000008	00	Char	A class in which the user is allowed to define profiles. (The user has the CLAUTH attribute.) The user can also define profiles in any other classes with POSIT values matching these classes.
CONGRPCT	043	10	80	00000004	00	Int	The number of groups that the user is connected to.
CONGRPNM	044	80	80	00000008	00	Char	A group that the user is connected to.

User template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
USRCNT USRNM USRDATA USRFLG	045 046 047 048	10 80 80 A0	00 80 80 80	00000004 00000008 00000000 00000001	00 00 00 00	Int	Reserved for installation use. Note: Intended usage: For installation to store additional data in this profile. USRNM should have a field name to use as a key to identify each unique occurrence of a row in the repeat group. USRDATA and USRFLG hold the data associated with that name. For more information, see <i>Example 5: Updating the installation fields</i> , in <i>Examples of ICHEINTY, ICHECTEST, and ICHEACTN macro usage in z/OS Security Server RACF Macros and Interfaces</i> .
SECLABEL	049	00	80	00000008	00	Char	Security label.
CGGRPCT	050	10	80	00000004	00	Int	Number of Connect Group entries. Information from the following CGxxx fields is also available through the logical connect profiles (ICHEINTY with CLASS=CONNECT) in the database. See “Connect template for the RACF database” on page 629 for more details.
CGGRPNM	051	82	80	00000008	00	Char	Connect Group Entry Name.
CGAUTHDA	052	80	A0	00000003	FF	Date	Date the user was connected.
CGAUTHOR	053	80	80	00000008	FF	Char	Owner of connect occurrence.
CGLJTIME	054	81	00	00000004	FF	Time	Time of RACROUTE REQUEST=VERIFY.
CGLJDATE	055	81	20	00000003	FF	Date	Date of RACROUTE REQUEST=VERIFY.
CGUACC	056	A0	80	00000001	00	Bin	Default universal access.
CGINITCT	057	81	00	00000002	FF	Int	Number of RACROUTE REQUEST=VERIFY requests that were successfully processed where the value specified in the CGRPNM field was the current connect group.
CGFLAG1	058	A0	80	00000001	00	Bin	If bit 0 is on, the user has the ADSP attribute in that group.
CGFLAG2	059	A0	80	00000001	00	Bin	If bit 0 is on, the user has the SPECIAL attribute in that group.
CGFLAG3	060	A0	80	00000001	00	Bin	If bit 0 is on, the user has the OPERATIONS attribute in that group.
CGFLAG4	061	A0	80	00000001	00	Bin	If bit 0 is on, the user has the REVOKE attribute in that group.
CGFLAG5	062	A0	80	00000001	00	Bin	If bit 0 is on, the user has the GRPACC attribute in that group.
CGNOTUAC	063	A0	80	00000001	00	Bin	If bit 0 is on, the user must be specifically authorized (by the PERMIT command) to use a terminal. If off, RACF uses the terminal's UACC.
CGGRPAUD	064	A0	80	00000001	00	Bin	If bit 0 is on, the user has the GROUP AUDITOR attribute in that group.
CGREVKDT	065	80	20	00000000	00	Date	The date the user is revoked. This field either has length 0, or contains a 3-byte revoke date.
CGRESMDT	066	80	20	00000000	00	Date	The date the user is resumed. This field either has length 0, or contains a 3-byte resume date.
TUCNT	067	10	00	00000002	00	Int	Number of user ID associations.

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
TUKEY	068	80	00	00000016	00	Char	Associated node and user ID. Byte Meaning when set 0–7 The associated node name. 8–15 The associated user ID.
TUDATA	069	80	00	00000000			Associated user ID association data Byte Meaning when set 0 Version number of the TUDATA entry.
						Bin	1 Bitstring 0 Specifies the user as having (bit is on) or not having (bit is off) a peer user ID association. 1 Specifies the user as being (bit is on) the manager of a managed user ID association. 2 Specifies the user as being (bit is on) managed by a managed user ID association. 3 An association request for this user is pending (bit is on) on a remote RRSF node. 4 An association request for this user is pending (bit is on) on the local RRSF node. 5 Specifies that password synchronization is in effect (bit is on) for this peer-user ID association. 6 Specifies that the association request for this user was rejected (bit is on). 7 Reserved for IBM's use.
							2–20 Reserved for IBM's use.
						Date	2–24 The date the user ID association was defined. (yyyymmdd)

User template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
						Time	25–32 The time the user ID association was defined. For the format of the time, see the TIME macro as documented in z/OS MVS Programming: Assembler Services Reference IAR-XCT .
						Char	32–36 The date the user ID association was approved or refused. (yyyymmdd)
						Int	37–44 The time the user ID association was approved or refused. For the format of the time, see the TIME macro as documented in z/OS MVS Programming: Assembler Services Reference IAR-XCT .
							45–56 Reserved for IBM's use.
						Char	57–64 The user ID that created the entry.
CERTCT	070	10	00	00000004	00		Number of certificate names.
CERTNAME	071	80	00	00000000	00		Name of certificate. Names correspond to profiles in the DIGTCERT class for the user.
CERTLABL	072	80	00	00000000	00		Label associated with the certificate.
CERTSJDN	073	80	00	00000000	00		Subject's distinguished name.
CERTPUBK	074	80	00	00000000	00		Public key associated with the certificate.
CERTRSV3	075	80	00	00000000	00		Reserved for IBM's use.
FLAG9	076	20	80	00000001	00		Restricted Access = BIT0.
NMAPCT	077	10	00	00000004	00		Number of DIGTNMAP Mapping Profiles that specify this user ID.
NMAPLABL	078	80	00	00000000	00		Label associated with this mapping.
NMAPNAME	079	80	00	00000000	00		Name of mapping profile. The names correspond to profiles in the DIGTNMAP class.
NMAPRSV1	080	80	00	00000000	00		Reserved for IBM's use.
NMAPRSV2	081	80	00	00000000	00		Reserved for IBM's use.
NMAPRSV3	082	80	00	00000000	00		Reserved for IBM's use.
NMAPRSV4	083	80	00	00000000	00		Reserved for IBM's use.
NMAPRSV5	084	80	00	00000000	00		Reserved for IBM's use.
PWDENV	085	00	08	00000000	00	Bin	Internal form of the enveloped RACF password.
PASSASIS	086	20	80	00000001	00	Bin	Identifies the user as having (bit 0 is on) or not having used a mixed case password.
PHRASE	087	04	80	00000000	FF	BIN	The password phrase associated with this user.
PHRDATE	088	00	A0	00000003	FF	BIN	The date the Pass Phrase was last changed.
PHRGEN	089	00	00	00000001	FF	INT	Current password phrase generation number.

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
PHRCNT	090	10	00	00000004	00	INT	Number of old password phrases.
OLDPHRNM	091	80	00	00000001	00	INT	Generation number of password phrase.
OLDPHR	092	84	00	00000008	FF	BIN	Previous password phrase, truncated to 8 bytes.
CERTSEQN	093	00	00	00000004	00	INT	Sequence number that is incremented whenever a certificate for the user is added, deleted, or altered.
PPHENV	094	00	00	00000000	00	BIN	Internal form of the enveloped RACF password phrase.
DMAPCT	095	10	00	00000004	00		Number of IDIDMAP Mapping Profiles that specify this user ID.
DMAPLABL	096	80	00	00000000	00		Label associated with this mapping.
DMAPNAME	097	80	00	00000000	00		Name of mapping profile. The names correspond to profiles in the IDIDMAP class.
DMAPRSV1	098	80	00	00000000	00		Reserved for IBM's use.
DMAPRSV2	099	80	00	00000000	00		Reserved for IBM's use.
PWDX	100	04	80	00000000	00		Password extension
OPWDXCT	101	10	00	00000004	00		Password history extension: count of entries
OPWDXGEN	102	80	00	00000001	FF		Password history extension: generation number
OPWDX	103	84	00	00000000	00		Password history extension: password value
PHRASEX	104	04	00	00000000	00		Password phrase extension
PHRCNTX	105	10	00	00000004	00		Phrase history extension: count of entries
OLDPHRNX	106	80	00	00000001	FF		Phrase history extension: generation number
OLDPHRX	107	84	00	00000000	00		Phrase history extension: phrase value
FLAGROA	108	20	80	00000001	00		ROAUDIT - Bit
MFAFLBK	109	20	80	00000001	00		User can fall back to password logon
FACTORN	110	10	80	00000004	00		Number of defined factors
FACTOR	111	80	80	00000000	00		Factor name - repeat
FACACDT	112	82	80	00000008	FF		Factor active-on date - repeat
FACTAGS	113	80	00	00000000	00		Factor configuration data - repeat
MFAPOLN	120	10	80	00000004	00		Number of defined policies
MFAPOLNM	121	80	80	00000000	00		Policy name - repeat
PHRINT	122	00	80	00000002	00	INT	The password change interval in days (represented by a number between 0 and 65534) that the user's password phrase is in effect. If it is 65535 (X'FFFF'), the user's password phrase never expires. See the description of the SETR PASSWORD (PHRASEINT...) processing instructions in <i>z/OS Security Server RACF Command Language Reference</i> for more details.

Field name	Field ID	Flag 1	Flag 2	Combination field IDs					Type	
The following are the COMBINATION fields of the USER template.										
DEFDATE	000	40	00	004	000	000	000	000		Combination.

User template

Field name	Field ID	Flag 1	Flag 2	Combination field IDs					Type	
CREADATE	000	40	00	004	000	000	000	000		Fields.
OWNER	000	40	00	005	000	000	000	000		
PASSDATA	000	40	00	012	013	000	000	000		
NAME	000	40	00	014	000	000	000	000		
OLDPSWDS	000	40	00	026	027	000	000	000		
LOGINFO	000	40	00	035	036	000	000	000		
FIELD	000	40	00	038	039	040	000	000		
USERDATA	000	40	00	046	047	048	000	000		
CGDEFDAT	000	40	00	052	000	000	000	000		
CGCREADT	000	40	00	052	000	000	000	000		
CGOWNER	000	40	00	053	000	000	000	000		
TUENTRY	000	40	00	068	069	000	000	000		
CERTLIST	000	40	00	071	072	000	000	000		
CERTLST2	000	40	00	071	072	073	074	000		
CERTLST3	000	40	00	071	072	073	000	000		
CERTSIGL	000	40	00	071	073	074	000	000		
OLDPHRES	000	40	00	091	092	000	000	000		
DMAPLST1	000	40	00	096	097	000	000	000		Combination for distributed identity.
OLDPWDX	000	40	00	102	103	000	000	000		Alias for extended pwd history entry
OLDPHREX	000	40	00	106	107	000	000	000		Alias for extended phrase history entry
FACINFO	000	40	00	111	112	113	000	000		MFA factor repeat group entry

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
The following is the DFP segment of the USER template.							
DFP	001	00	00	00000000	00		Start of segment fields
DATAAPPL	002	00	00	00000000	00	Char	Data Application; maximum length = 8
DATACLAS	003	00	00	00000000	00	Char	Data Class; maximum length = 8
MGMTCLAS	004	00	00	00000000	00	Char	Management Class; maximum length = 8
STORCLAS	005	00	00	00000000	00	Char	Storage Class; maximum length = 8
The following is the TSO segment of the USER template.							
TSO	001	00	00	00000000	00		Start of segment fields
TACCNT	002	00	00	00000000	00	Char	Default account numbers; maximum length = 40
TCOMMAND	003	00	00	00000000	00	Char	Default command at logon; maximum length = 80
TDEST	004	00	00	00000000	00	Char	Destination identifier; maximum length = 8
THCLASS	005	00	00	00000000	00	Char	Default hold class; maximum length = 1
TJCLASS	006	00	00	00000000	00	Char	Default job class
TLPROC	007	00	00	00000000	00	Char	Default logon procedure; maximum length = 8
TLSIZE	008	00	00	00000004	00	Int	Logon size

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
TMCLASS	009	00	00	00000000	00	Char	Default message class; maximum length = 1
TMSIZE	010	00	00	00000004	00	Int	Maximum region size
TOPTION	011	20	00	00000001	00	Bin	Default for mail notices and OIDcard
TPERFORM	012	00	00	00000004	00	Int	Performance group; stored as a two-byte value
TRBA	013	00	00	00000003	00	Bin	RBA of user's broadcast area
TSCLASS	014	00	00	00000000	00	Char	Default sysout class
TUDATA	015	00	00	00000002	00	Bin	2 bytes of hex user data
TUNIT	016	00	00	00000000	00	Char	Default unit name; maximum length = 8
TUPT	017	00	00	00000000	00	Bin	Data from UPT control block
TSOSLABL	018	00	00	00000000	00	Char	Default logon SECLABEL; maximum length = 8
TCONS	019	00	00	00000000	00	Char	Consoles support
The following is the CICS segment of the USER template.							
CICS	001	00	00	00000000	00		Start of segment fields
OPIDENT	002	00	00	00000003	00	Char	Operator identification; 1 to 3 bytes in length
OPCLASSN	003	10	00	00000004	00	Int	Count of operator class values
OPCLASS	004	80	00	00000001	00	Int	Operator class
OPPRTY	005	00	40	00000002	00	Int	Operator priority
XRFSOFF	006	20	00	00000001	00	Bin	XRF re-signon option: <ul style="list-style-type: none"> • Bit 0 on = FORCE • Bit 0 off = NOFORCE
TIMEOUT	007	00	40	00000002	00	Bin	Terminal timeout value Two 1-byte binary fields: <ul style="list-style-type: none"> • First byte = hours (0–99) • Second byte = minutes (0–59) Special case: The following examples are handled the same way: <ul style="list-style-type: none"> • First byte = 0 hours • Second byte = 60 minutes • First byte = 1 hours • Second byte = 0 minutes
RSLKEYN	008	10	00	00000004	00	Int	Count of resource security level (RSL) key values
RSLKEY	009	80	00	00000002	00	Int	RSL key value
TSLKEYN	010	10	00	00000004	00	Int	Count of transaction security level (TSL) key values
TSLKEY	011	80	00	00000002	00	Int	TSL key value
The following is the LANGUAGE segment of the USER template.							
LANGUAGE	001	00	00	00000000	00		Start of segment fields
USERNL1	002	00	80	00000003	00	Char	User's primary language; 3-character code returned by the MVS message service. For more information, see z/OS MVS Programming: Assembler Services Guide .
USERNL2	003	00	80	00000003	00	Char	User's secondary language

User template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
The following is the OPERPARM segment of the USER template.							
OPERPARM	001	00	00	00000000	00		Start of segment fields
OPERSTOR	002	00	00	00000002	00	Bin	STORAGE keyword
OPERAUTH	003	00	00	00000002	00	Bin	AUTH keyword: <ul style="list-style-type: none">• X'8000' = MASTER• X'4000' = ALL• X'2000' = SYS• X'10000' = IO• X'0800' = CONS• X'0400' = INFO
OPERMFRM	004	00	00	00000002	00	Bin	MFORM keyword: <ul style="list-style-type: none">• Bit 0 indicates T• Bit 1 indicates S• Bit 2 indicates J• Bit 3 indicates M• Bit 4 indicates X
OPERLEVL	005	00	00	00000002	00	Bin	LEVEL keyword: <ul style="list-style-type: none">• Bit 0 indicates R• Bit 1 indicates I• Bit 2 indicates CE• Bit 3 indicates E• Bit 4 indicates IN• Bit 5 indicates NB• Bit 6 indicates ALL Bit 6 is mutually exclusive with all other bits except Bit 5.
OPERMON	006	00	00	00000002	00	Bin	MONITOR keyword: <ul style="list-style-type: none">• Bit 0 indicates JOBNAMEs• Bit 1 indicates JOBNAMEsT• Bit 2 indicates SESS• Bit 3 indicates SESST• Bit 4 indicates STATUS Bits 0 and 1 are mutually exclusive, as are bits 2 and 3.
OPERROUT	007	00	00	00000000	00	Bin	ROUTCODE keyword; 16-bit length bitstring in which each bit indicates a particular ROUTCODE.
OPERLOGC	008	00	00	00000001	00	Bin	LOGCMDRESP keyword. Value Meaning when set X'80' Indicates SYSTEM was specified. X'40' Indicates NO was specified.

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
OPERMGID	009	00	00	00000001	00	Bin	MIGID keyword. Value Meaning when set X'80' Indicates YES was specified. X'40' Indicates NO was specified.
OPERDOM	010	00	00	00000001	00	Bin	DOM keyword. Value Meaning when set X'80' Indicates NORMAL was specified. X'40' Indicates ALL was specified. X'20' Indicates NONE was specified.
OPERKEY	011	00	00	00000000	00	Bin	KEY keyword; maximum length = 8
OPERCMD5	012	00	00	00000000	00	Bin	CMD5 keyword; maximum length = 8 (or '*')
OPERUD	013	00	00	00000001	00	Bin	UD keyword. Value Meaning when set X'80' Indicates YES was specified. X'40' Indicates NO was specified.
OPERM CNT	014	10	00	00000004	00	Bin	Count of MSCOPE systems
OPERMSCP	015	80	00	00000008	00	Bin	MSCOPE systems
OPERALTG	016	00	00	00000000	00	Bin	ALTGRP keyword Value Meaning when set X'80' Indicates YES was specified. X'40' Indicates NO was specified.
OPERAUTO	017	00	00	00000001	00	Bin	AUTO keyword; X'80' indicates YES; X'40' indicates NO.
OPERHC	018	00	00	00000001	00	BIN	HC keyword; X'80' indicates YES; X'40' indicates NO.
OPERINT	019	00	00	00000001	00	BIN	INTIDS keyword; X'80' indicates YES; X'40' indicates NO.
OPERUNKN	020	00	00	00000001	00	BIN	UNKNIDS keyword; X'80' indicates YES; X'40' indicates NO.
The following is the WORK ATTRIBUTES segment of the USER template.							
WORKATTR	001	00	80	00000000	00		Start of segment fields
WANAME	002	00	80	00000000	00	Char	User name for SYSOUT; maximum length = 60
WABLDG	003	00	80	00000000	00	Char	Building for delivery; maximum length = 60
WADEPT	004	00	80	00000000	00	Char	Department for delivery; maximum length = 60

User template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
WAROOM	005	00	80	00000000	00	Char	Room for delivery; maximum length = 60
WAADDR1	006	00	80	00000000	00	Char	SYSOUT address line 1; maximum length = 60
WAADDR2	007	00	80	00000000	00	Char	SYSOUT address line 2; maximum length = 60
WAADDR3	008	00	80	00000000	00	Char	SYSOUT address line 3; maximum length = 60
WAADDR4	009	00	80	00000000	00	Char	SYSOUT address line 4; maximum length = 60
WAACCNT	010	00	80	00000000	00	Char	Account number; maximum length = 255
WAEMAIL	011	00	94	00000000	00	Char	E-mail address; maximum length = 246
The following is the OMVS segment of the USER template.							
OMVS	001	00	00	00000000	00		Start of segment fields
UID	002	00	10	00000004	FF	Int	UID
HOME	004	00	00	00000000	00	Char	HOME Path; maximum length = 1023
PROGRAM	005	00	00	00000000	00	Char	Initial Program; maximum length = 1023
CPUTIME	006	00	00	00000004	FF	Int	CPUTIMEMAX
ASSIZE	007	00	00	00000004	FF	Int	ASSIZEMAX
FILEPROC	008	00	00	00000004	FF	Int	FILEPROCMAX
PROCUSER	009	00	00	00000004	FF	Int	PROCUSERMAX
THREADS	010	00	00	00000004	FF	Int	THREADSMAX
MMAPAREA	011	00	00	00000004	FF	Int	MMAPAREAMAX
MEMLIMIT	012	00	00	00000000	0	Char	MEMLIMIT; maximum length = 9
SHMEMMAX	013	00	00	00000000	0	Char	SHMEMMAX; maximum length = 9
The following is the NETVIEW segment of the USER template.							
NETVIEW	001	00	00	00000000	00		Start of segment fields
IC	002	00	00	00000000	00	Char	The command or command list to be processed by NetView for this operator when the operator logs on to Netview; maximum length = 255
CONSNAME	003	00	00	00000000	00	Char	The default MCS console identifier; maximum length = 8
CTL	004	20	00	00000001	00	Bin	<p>CTL keyword – Specifies whether a security check is performed for this NetView operator when they try to use a span or try to do a cross-domain logon.</p> <p>Value Meaning when set</p> <p>X'00' Indicates CTL was not specified or CTL(SPECIFIC) was specified.</p> <p>X'80' Indicates CTL(GLOBAL) was specified.</p> <p>X'40' Indicates CTL(GENERAL) was specified.</p>

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
MSGRECV	005	20	00	00000001	00	Bin	MSGRECV keyword Value Meaning when set X'00' Indicates the operator can receive unsolicited messages that are not routed to a specific NetView operator. X'80' Indicates the operator cannot receive unsolicited messages that are not routed to a specific NetView operator.
OPCLASSN	006	10	00	00000004	00	Int	Count of operator class values.
OPCLASS	007	80	40	00000002	00	Int	Specifies a NetView scope class for which the operator has authority. This is a 2-byte repeating field. Each member can have fixed-binary values from 1 to 2040.
DOMAINSN	008	10	00	00000004	00	Int	The number of domains the NetView operator controls.
DOMAINS	009	80	00	00000000	00	Char	Specifies the identifier of NetView programs in another NetView domain for which this operator has authority. This is a variable length (5-character maximum) repeating field.
NGMFADMN	010	20	00	00000001	00	Bin	NGMFADMN keyword Value Meaning when set X'00' The NetView operator does not have administrator authority to the NetView Graphic Monitor Facility (NGMF). X'80' The NetView operator has administrator authority to the NetView graphic monitor facility (NGMF).
NGMFVSPN	011	00	00	00000000	00		NetView Graphic Monitor Facility view span options; maximum length = 8
The following is the DCE segment of the USER template.							
DCE	001	00	00	00000000	00		Start of segment fields
UUID	002	00	00	00000036	FF	Char	User's DCE principal's UUID; exactly 36 characters, in the format <i>nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnn</i> where <i>n</i> is any hexadecimal digit.
DCENAME	003	00	00	00000000	00	Char	User's DCE principal name; maximum length = 1023
HOMECELL	004	00	00	00000000	00	Char	Home cell for this DCE user; maximum length = 1023, and it must start with either <i>/.../</i> or <i>././</i>
HOMEUUID	005	00	00	00000036	FF	Char	Home cell UUID; exactly 36 characters, in the format <i>nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnn</i> where <i>n</i> is any hexadecimal digit.
DCEFLAGS	006	20	00	00000001	00	Bin	User flags
DPASSWDS	007	00	00	00000000	00	Char	Current DCE password
DCEENCRY	008	00	00	00000071	00	Bin	PW mask/encrypt key

User template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
The following is the OVM segment of the USER template.							
OVM	001	00	00	00000000	00		Start of segment fields
UID	002	00	00	00000004	FF	Int	OVM - UID
HOME	003	00	00	00000000	00	Char	Home path; maximum length = 1023
PROGRAM	004	00	00	00000000	00	Char	Initial program; maximum length = 1023
FSROOT	005	00	00	00000000	00	Char	File system root; maximum length = 1023
The following is the LNOTES segment of the USER template.							
LNOTES	001	00	00	00000000	00		Start of segment fields
SNAME	002	00	14	00000000	00	Char	User's short name; maximum length = 64
The following is the NDS segment of the USER template.							
NDS	001	00	00	00000000	00		Start of segment fields
UNAME	002	00	14	00000000	00	Char	User's user name; maximum length = 246
The following is the KERB segment of the USER template.							
KERB	001	00	00	00000000	00		Start of segment fields
KERBNAME	002	00	00	00000000	00	Char	Kerberos principal name
MINTKTLF	003	00	00	00000000	00	Char	Reserved for IBM's use.
MAXTKTLF	004	00	00	00000000	00	Char	Maximum ticket life
DEFTKTLF	005	00	00	00000000	00	Char	Reserved for IBM's use.
SALT	006	00	00	00000000	00	Char	Current key salt
ENCTYPE	007	00	00	00000000	00	Char	Encryption type
CURKEYV	008	00	00	00000000	00	Char	Current key version
CURKEY	009	00	00	00000000	00	Char	Current key value
PREVKEYV	010	00	00	00000000	00	Char	Previous key version
PREVKEY	011	00	00	00000000	00	Char	Previous key value
ENCRYPT	012	00	00	00000004	55	Bin	Encryption type
KEYFROM	013	00	00	00000000	00	Char	Key source 0 = PASSWORD 1 = PHRASE
The following is the PROXY segment of the USER template.							
PROXY	001	00	00	00000000	00		Start of segment fields
LDAPHOST	002	00	00	00000000	00	Char	LDAP server URL; maximum length: 1023
BINDDN	003	00	00	00000000	00	Char	Bind distinguished name; maximum length: 1023
BINDPW	004	00	08	00000000	00	Char	Bind password; maximum length: 128
BINDPWKY	005	00	08	00000071	00	Char	Bind password mask or encrypt key
The following is the EIM segment of the USER template.							
EIM	001	00	00	00000000	00	Char	Start of segment fields
LDAPPROF	002	00	00	00000000	00	Char	LDAPBIND profile name
The following is the CSDATA segment of the USER template.							

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
CSDATA	001	00	00	0	0		Start of segment fields for custom fields Note: Intended usage for these fields is dictated by your installation. See z/OS Security Server RACF Security Administrator's Guide for more information on custom fields.
CSCNT	002	10	00	4	00	Integer	Count of custom fields
CSTYPE	003	80	00	1	01	Bin	Custom field type: <ul style="list-style-type: none"> • 01 - character • 02 - numeric • 03 - flag • 04 - hex
CSKEY	004	80	00	00	00	Char	Custom field keyword; maximum length = 8
CSVALUE	005	80	00	0	00	Char	Custom field value

Field name	Field ID	Flag 1	Flag 2	Combination field IDs					Type	
The following is a COMBINATION field of the CSDATA segment of the USER template.										
CSCDATA	000	40	00	003	004	005	000	000	Char	Combination field for custom fields

Connect template for the RACF database

The connect template is included to maintain compatibility with previous releases. You can continue to code macros to manipulate CONNECT data. This template is provided to show what fields continue to be supported. Information that was formerly stored in CONNECT profiles was moved to the USER profile. The information is in the CGGRPCT repeat group, and the fields are prefixed by "CG".

Note:

1. Application developers should not depend on being able to use RACROUTE REQUEST=EXTRACT for the BASE segment fields on any security product other than RACF. These products are expected to support only such segments as DFP and TSO.
2. The default values for this template are in the user template.

The contents of the connect template are as follows:

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
CONNECT	001	00	00	00000000			
ENTYPE	002	00	00	00000001		Int	The number (3) corresponding to connect profiles.
VERSION	003	00	00	00000001		Int	The version field from the profile. Always X'01'.
AUTHDATE	004	00	A0	00000003		Date	The date the user was connected to the group.
AUTHOR	005	00	80	00000008		Char	The owner (user ID) of the connect entry.
LJTIME	006	01	00	00000004		Time	The last recorded time that RACROUTE REQUEST=VERIFY was last issued for this user and group.

Connect template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
LJDATE	007	01	20	00000003		Date	The last recorded date that RACROUTE REQUEST=VERIFY was last issued for this user and group.
UACC	008	20	80	00000001		Bin	<p>The default universal access authority assigned to the user for this group.</p> <p>Bit</p> <p>Meaning when set</p> <p>0 ALTER access</p> <p>1 CONTROL access</p> <p>2 UPDATE access</p> <p>3 READ access</p> <p>4 EXECUTE access</p> <p>5–6 Reserved for IBM's use</p> <p>7 EXECUTE access</p>
INITCNT	009	01	00	00000002		Int	The number of RACROUTE REQUEST=VERIFY macro instructions issued for this user and group.
FLAG1	010	20	80	00000001		Bin	Identifies the user as having (bit 0 is on) or not having the ADSP attribute.
FLAG2	011	20	80	00000001		Bin	Identifies the user as having (bit 0 is on) or not having the group-SPECIAL attribute.
FLAG3	012	20	80	00000001		Bin	Identifies the user as having (bit 0 is on) or not having the group-OPERATIONS attribute.
FLAG4	013	20	80	00000001		Bin	Identifies the user as having (bit 0 is on) or not having the REVOKE attribute.
FLAG5	014	20	80	00000001		Bin	Identifies the user as having (bit 0 is on) or not having the GRPACC attribute.
NOTRMUAC	015	20	80	00000001		Bin	Identifies whether the user must be authorized by the PERMIT command with at least READ authority to access a terminal. (If not, RACF uses the terminal's universal access authority.) If bit 0 is on, the user must be specifically authorized to use the terminal.
GRPAUDIT	016	20	80	00000001		Bin	Identifies the user as having (bit 0 is on) or not having the group-AUDITOR attribute.
REVOKEDT	017	00	20	00000000		Date	The date the user is revoked. This field either has length 0, or contains a 3-byte revoke date.
RESUMEDT	018	00	20	00000000		Date	The date the user is resumed. This field either has length 0, or contains a 3-byte resume date.

Field name	Field ID	Flag 1	Flag 2	Combination field IDs					Type	
The following are the COMBINATION fields of the CONNECT template.										
DEFDATE	000	40	00	004	000	000	000	000	Char	Combination.

Field name	Field ID	Flag 1	Flag 2	Combination field IDs					Type	
CREADATE	000	40	00	004	000	000	000	000	Char	Fields.
OWNER	000	40	00	005	000	000	000	000	Char	

Data set template for the RACF database

The data set template describes the fields of the data set profiles in a RACF database.

NOT Programming Interface Information			
ACL2VAR AUDITQF AUDITQS	CATEGORY FIELD FLDCNT	FLDFLAG FLDNAME	FLDVALUE NUMCTGY
End NOT Programming Interface Information			

Note:

1. Application developers should not depend on being able to use RACROUTE REQUEST=EXTRACT for the BASE segment fields on any security product other than RACF. These products are expected to support only such segments as DFP and TSO.
2. The TME segment fields are intended to be updated by Tivoli applications, which manage updates, permissions, and cross-references among the fields. The TME fields should only be directly updated on an exception basis. See *z/OS Security Server RACF Command Language Reference* for formats of the field data as enforced by the RACF commands. Use caution when directly updating TME fields, as the updates might be overridden by subsequent actions of Tivoli applications.

The contents of the data set template are as follows:

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
DATASET	001	00	00	00000000	00		
ENTYPE	002	00	00	00000001	04	Int	The number (4) corresponding to data set profiles.
VERSION	003	00	00	00000001	01	Int	The version field from the profile. Always X'01'.
CREADATE	004	00	20	00000003	FF	Date	The date the data set was initially defined to RACF; 3-byte date.
AUTHOR	005	00	00	00000008	FF	Char	The owner of the data set.
LREFDAT	006	01	20	00000003	FF	Date	The date the data set was last referenced; 3-byte date.
LCHGDAT	007	01	20	00000003	FF	Date	The date the data set was last updated; 3-byte date.
ACCSALTR	008	01	00	00000002	FF	Int	The number of times the data set was accessed with ALTER authority.
ACSCNTL	009	01	00	00000002	FF	Int	The number of times the data set was accessed with CONTROL authority.
ACSUPDT	010	01	00	00000002	FF	Int	The number of times the data set was accessed with UPDATE authority.
ACSREAD	011	01	00	00000002	FF	Int	The number of times the data set was accessed with READ authority.

Data set template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
UNIVACS	012	20	00	00000001	00	Bin	<p>The universal access authority for the data set.</p> <p>Bit</p> <p>Meaning when set</p> <p>0 ALTER access</p> <p>1 CONTROL access</p> <p>2 UPDATE access</p> <p>3 READ access</p> <p>4 EXECUTE access</p> <p>5–6 Reserved for IBM's use</p> <p>7 NONE access</p>
FLAG1	013	20	00	00000001	00	Bin	<p>Identifies whether the data set is a group data set. If bit 0 is on, the data set is a group data set.</p>
AUDIT	014	20	00	00000001	00	Bin	<p>Audit Flags.</p> <p>Bit</p> <p>Meaning when set</p> <p>0 Audit all accesses</p> <p>1 Audit successful accesses</p> <p>2 Audit accesses that fail</p> <p>3 No auditing</p> <p>4–7 Reserved for IBM's use</p>
GROUPNM	015	00	00	00000008	FF	Char	<p>The current connect group of the user who created this data set.</p>
DSTYPE	016	20	00	00000001	00	Bin	<p>Identifies the data set as a VSAM, non-VSAM (or generic), MODEL or TAPE data set.</p> <p>Bit</p> <p>Meaning when set</p> <p>0 VSAM data set (non-VSAM if this bit is set to 0)</p> <p>1 MODEL profile</p> <p>2 Type = TAPE when set on</p> <p>3–7 Reserved for IBM's use</p>
LEVEL	017	00	00	00000001	FF	Int	<p>Data set level.</p>
DEVTYPE	018	00	00	00000004	FF	Bin	<p>The type of device on which the data set resides; only for non-model, discrete data sets.</p>

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
DEVTPPX	019	00	00	00000008	FF	Char	The EBCDIC name of the device type on which the data set resides; only for non-model, discrete data sets.
GAUDIT	020	20	00	00000001	00	Bin	Global audit flags. (Audit options specified by a user with the AUDITOR or group-AUDITOR attribute.) Bit Meaning when set 0 Audit all accesses 1 Audit successful accesses 2 Audit accesses that fail 3 No auditing 4–7 Reserved for IBM's use
INSTDATA	021	00	00	00000000	00	Char	Installation data; maximum length = 255.
GAUDITQF	025	00	00	00000001	FF	Bin	Global audit FAILURES qualifier. The AUDITQS, AUDITQF, GAUDITQS, and GAUDITQF fields have the following format: Value Meaning when set X'00' Log access at READ level X'01' Log access at UPDATE level X'02' Log access at CONTROL level X'03' Log access at ALTER level
AUDITQS	022	00	00	00000001	FF	Bin	Audit SUCCESS qualifier.
AUDITQF	023	00	00	00000001	FF	Bin	Audit FAILURES qualifier.
GAUDITQS	024	00	00	00000001	FF	Bin	Global audit SUCCESS qualifier.
WARNING	026	20	00	00000001	00	Bin	Identifies the data set as having (bit 7 is on) or not having the WARNING attribute.
SECLEVEL	027	00	00	00000001	FF	Int	Data set security level.
NUMCTGY	028	10	00	00000004	00	Int	The number of categories.
CATEGORY	029	80	00	00000002	00	Bin	A list of numbers corresponding to the categories to which this data set belongs.
NOTIFY	030	00	00	00000000	00	Char	User to be notified when access violations occur against a data set protected by this profile.
RETPD	031	00	00	00000000	00	Int	The number of days protection is provided for the data set. If used, the field is a two-byte binary number.
ACL2CNT	032	10	00	00000004	00	Int	The number of program and user combinations currently authorized to access the data set.

Data set template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
PROGRAM	033	80	00	00000008	00	Char	The name of a program currently authorized to access the data set, or a 1-byte flag followed by 7 bytes reserved for IBM's use.
USER2ACS	034	80	00	00000008	00	Char	User ID or group.
PROGACS	035	80	00	00000001	00	Bin	The access authority of the program and user combinations.
PACSCNT	036	80	00	00000002	00	Int	Access count.
ACL2VAR	037	80	00	00000000	00	Char	Additional conditional data, 9-byte length, in which the first byte tells what type of access is allowed and the remaining 8 bytes contain the data.
FLDCNT	038	10	00	00000004	00		Reserved for IBM's use.
FLDNAME	039	80	00	00000008	00		Reserved for IBM's use.
FLDVALUE	040	80	00	00000000	00		Reserved for IBM's use.
FLDFLAG	041	A0	00	00000001	00		Reserved for IBM's use.
VOLCNT	042	10	00	00000004	00	Int	The number of volumes containing the data set.
VOLSER	043	80	00	00000006	00	Char	A list of the serial numbers of the volumes containing the data set.
ACLCNT	044	10	00	00000004	00	Int	The number of users and groups currently authorized to access the data set.
USERID	045	80	00	00000008	00	Char	The user ID or group name of each user or group authorized to access the data set.
USERACS	046	A0	00	00000001	00	Bin	<p>The access authority that each user or group has for the data set.</p> <p>Bit</p> <p>Meaning when set</p> <p>0 ALTER access</p> <p>1 CONTROL access</p> <p>2 UPDATE access</p> <p>3 READ access</p> <p>4 EXECUTE access</p> <p>5–6 Reserved for IBM's use</p> <p>7 NONE access</p>
ACSCNT	047	80	00	00000002	00	Int	The number of times the data set was accessed by each user or group.
USRCNT	048	10	00	00000004	00	Int	Reserved for installation use.
USRNM	049	80	00	00000008	00		Reserved for installation use.
USRDATA	050	80	00	00000000	00		Reserved for installation use.
USRFLG	051	A0	00	00000001	00		Reserved for installation use.
SECLABEL	052	00	00	00000008	00	Char	Security label.

Field name	Field ID	Flag 1	Flag 2	Combination field IDs					Type	
The following are the COMBINATION fields of the data set template.										
DEFDATE	000	40	00	004	000	000	000	000	Char	Combination.
AUTHDATE	000	40	00	004	000	000	000	000	Char	Fields.
OWNER	000	40	00	005	000	000	000	000	Char	
UACC	000	40	00	012	000	000	000	000		
ACL2	000	40	00	033	034	035	036	037		
ACL2A3	000	40	00	033	034	035	037	000		
ACL2A2	000	40	00	033	034	035	036	000		
ACL2A1	000	40	00	033	034	035	000	000		
FIELD	000	40	00	039	040	041	000	000		
VOLUME	000	40	00	043	000	000	000	000		
ACL	000	40	00	045	046	047	000	000		
ACL1	000	40	00	045	046	000	000	000		
USERDATA	000	40	00	049	050	051	000	000		

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
The following is the DFP segment of the data set template.							
DFP	001	00	00	00000000	00		Start of segment fields
RESOWNER	002	00	00	00000008	FF	Char	Resource owner; must represent a user ID or group name
DATAKEY	003	00	00	00000000	00	Char	CKDS label of default key
ENCTYPES	004	00	00	00000004	00	DFP	Types of data set encrypted. Byte 1: X'80' - INTAPE X'40' - INPDSE X'20' - INSEQ X'08' - EXTAPE X'04' - EXPDSE X'02' - EXSEQ

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
The following is the TME segment of the data set template.							
TME	001	00	00	00000000	00		Start of segment fields
ROLEN	002	10	00	00000004	00	Int	Count of role-access specifications
ROLES	003	80	00	00000000	00	Char	Role-access specifications

General template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
The following is the CSDATA segment of the data set template.							
CSDATA	001	00	00	00000000	00		Start of segment fields for custom fields. Note: Intended usage for these fields is dictated by your installation. See the z/OS Security Server RACF Security Administrator's Guide for more information on custom fields.
CSCNT	002	10	00	00000004	00	Char	Count of custom fields
CSTYPE	003	80	00	00000001	01	Char	Custom field type <ul style="list-style-type: none"> • 01 - character • 02 - numeric • 03 - flag • 04 - hex
CSKEY	0004	80	00	00000000	00	Char	Custom field keyword
CSVALUE	005	80	00	00000000	00	Char	Custom field value

Field name	Field ID	Flag 1	Flag 2	Combination field IDs					Type	
The following is a COMBINATION field of the CSDATA segment of the data set template.										
CSCDATA	000	40	00	003	004	005	000	000	Char	Combination field for custom fields

General template for the RACF database

The general template describes the fields of general resource profiles in a RACF database.

NOT Programming Interface Information			
ACL2RSVD AUDITQF AUDITQS CATEGORY CURKEY CURKEYV	ENCTYPE FIELD FLDCNT FLDFLAG FLDNAME FLDVALUE	GAUDITQF GAUDITQS MEMCNT MEMLIST NUMCTGY PREVKEY	PREVKEYV RACLDSP RACLHDR SALT SSKEY
End NOT Programming Interface Information			

Note:

1. Application developers should not depend on being able to use RACROUTE REQUEST=EXTRACT for the BASE segment fields on any security product other than RACF. These products are expected to support only such segments as DFP and TSO.
2. The TME segment fields are intended to be updated by Tivoli applications, which manage updates, permissions, and cross-references among the fields. The TME fields should only be directly updated on an exception basis. See *z/OS Security Server RACF Command Language Reference* for formats of the field data as enforced by the RACF commands. Use caution when directly updating TME fields, as the updates might be overridden by subsequent actions of Tivoli applications.

The contents of the general template are as follows:

Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	Field being described
The following is the BASE segment of the GENERAL template.							
GENERAL	001	00	00	00000000	00		
ENTYPE	002	00	00	00000001	05	Int	The number (5) corresponding to profiles for resources defined in the class descriptor table.
VERSION	003	00	00	00000001	01	Int	The version field from the profile. Always X'01'.
CLASTYPE	004	00	00	00000001	FF	Int	The class to which the resource belongs (from the ID=class-number operand of the ICHERCDE macro).
DEFDATE	005	00	20	00000003	FF	Date	The date the resource was defined to RACF.
OWNER	006	00	00	00000008	FF	Char	The owner of the resource.
LREFDAT	007	01	20	00000003	FF	Date	The date the resource was last referenced.
LCHGDAT	008	01	20	00000003	FF	Date	The date the resource was last updated.
ACSALTR	009	01	00	00000002	FF	Int	The number of times the resource was accessed with ALTER authority.
ACSCNTL	010	01	00	00000002	FF	Int	The number of times the resource was accessed with CONTROL authority.
ACSUPDT	011	01	00	00000002	FF	Int	The number of times the resource was accessed with UPDATE authority.
ACSREAD	012	01	00	00000002	FF	Int	The number of times the resource was accessed with READ authority.
UACC	013	20	80	00000001	00	Bin	<p>The universal access authority for the resource.</p> <p>Bit</p> <p>Meaning when set</p> <p>0 ALTER access</p> <p>1 CONTROL access</p> <p>2 UPDATE access</p> <p>3 READ access</p> <p>4 EXECUTE access</p> <p>5–6 Reserved for IBM's use</p> <p>7 NONE access.</p>
AUDIT	014	20	00	00000001	00	Bin	<p>Audit flags.</p> <p>Bit</p> <p>Meaning when set</p> <p>0 Audit all accesses</p> <p>1 Audit successful accesses</p> <p>2 Audit accesses that fail</p> <p>3 No auditing</p> <p>4–7 Reserved for IBM's use</p>

General template

Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	Field being described
LEVEL	015	20	00	00000001	00	Int	Resource level.
GAUDIT	016	20	00	00000001	00	Bin	Global audit flags. Bit Meaning when set 0 Audit all accesses 1 Audit successful accesses 2 Audit accesses that fail 3 No auditing 4–7 Reserved for IBM's use
INSTDATA	017	00	00	00000000	00	Char	Installation data; maximum length = 255.
GAUDITQF	021	00	00	00000001	FF	Bin	Global audit FAILURES qualifier. Value Meaning X'00' Log access at READ authority X'01' Log access at UPDATE authority X'02' Log access at CONTROL authority X'03' Log access at ALTER authority
AUDITQS	018	00	00	00000001	FF	Bin	Audit SUCCESS qualifier. Value Meaning X'00' Log access at READ authority X'01' Log access at UPDATE authority X'02' Log access at CONTROL authority X'03' Log access at ALTER authority
AUDITQF	019	00	00	00000001	FF	Bin	Audit FAILURES qualifier. Value Meaning X'00' Log access at READ authority X'01' Log access at UPDATE authority X'02' Log access at CONTROL authority X'03' Log access at ALTER authority

Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	Field being described
GAUDITQS	020	00	00	00000001	FF	Bin	Global audit SUCCESS qualifier. Value Meaning X'00' Log access at READ authority X'01' Log access at UPDATE authority X'02' Log access at CONTROL authority X'03' Log access at ALTER authority
WARNING	022	20	00	00000001	00	Bin	Identifies the data set as having (bit 7 is on) or not having the WARNING attribute.
RESFLG	023	20	00	00000001	00	Bin	Resource profile flags: Bit Meaning when set 0 TAPEVOL can only contain one data set. 1 TAPEVOL profile is automatic. 2 Maintain TVTOC for TAPEVOL. 3–7 Reserved for IBM's use
TVTOCNT	024	10	00	00000004	00	Int	The number of TVTOC entries.
TVTOCSEQ	025	80	00	00000002	00	Int	The file sequence number of tape data set.
TVTOCCRD	026	80	20	00000003	00	Date	The date the data set was created.
TVTOCIND	027	A0	00	00000001	00	Bin	Data set profiles flag (RACF indicator bit): Bit Meaning when set 1 Discrete data set profile exists 2–7 Reserved for IBM's use
TVTOCDSN	028	80	00	00000000	00	Char	The RACF internal name.
TVTOCVOL	029	80	00	00000000	00	Char	This field is a list of the volumes on which the tape data set resides.
TVTOCRDS	030	80	00	00000000	00	Char	The name used when creating the tape data set; maximum length = 255.
NOTIFY	031	00	00	00000000	00	Char	The user to be notified when access violations occur against resource protected by this profile.
LOGDAYS	032	20	00	00000001	00	Bin	The days of the week the TERMINAL cannot be used. (Bit 0 equals Sunday, bit 1 equals Monday, and so on).
LOGTIME	033	00	00	00000000	00	Time	The time of the day the TERMINAL can be used.
LOGZONE	034	00	00	00000000	00	Bin	The time zone in which the terminal is located.
NUMCTGY	035	10	00	00000004	00	Int	Number of categories.
CATEGORY	036	80	00	00000002	00	Int	List of categories.
SECLEVEL	037	00	00	00000001	FF	Int	Resource security level.

General template

Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	Field being described
FLDCNT	038	10	00	00000004	00	Int	Reserved for IBM's use.
FLDNAME	039	80	00	00000008	00		Reserved for IBM's use.
FLDVALUE	040	80	00	00000000	00		Reserved for IBM's use.
FLDFLAG	041	A0	00	00000001	00		Reserved for IBM's use.
APPLDATA	042	00	00	00000000	00	Char	Application data.
MEMCNT	043	10	80	00000004	00	Int	The number of members.
MEMLST	044	80	80	00000000	00	Bin	The resource group member. For SECLABEL class, a 4-byte SMF ID.
VOLCNT	045	10	00	00000004	00	Int	Number of volumes in tape volume set.
VOLSER	046	80	00	00000006	00	Char	Volume serials of volumes in tape volume set.
ACLCNT	047	10	80	00000004	00	Int	The number of users and groups currently authorized to access the resource.
USERID	048	80	80	00000008	00	Char	The user ID or group name of each user or group authorized to access the resource.
USERACS	049	A0	80	00000001	00	Bin	<p>The access authority that each user or group has for the resource.</p> <p>Bit</p> <p>Meaning when set</p> <p>0 ALTER access</p> <p>1 CONTROL access</p> <p>2 UPDATE access</p> <p>3 READ access</p> <p>4 EXECUTE access</p> <p>5–6 Reserved for IBM's use</p> <p>7 NONE access</p> <p>Note: Each of the above access authority fields has mutually exclusive bits except for EXECUTE and NONE.</p>
ACSCNT	050	80	00	00000002	00	Int	The number of times the resource was accessed by each user or group.
USRCNT USRNM USRDATA USRFLG	051 052 053 054	10 80 80 A0	00 00 00 00	00000004 00000008 00000000 00000001	00 00 00 00	Int	Reserved for installation use. Reserved for installation use. Reserved for installation use. Reserved for installation use.
SECLABEL	055	00	00	00000008	00	Char	Security label.
ACL2CNT	056	10	00	00000004	00	Int	Number of entries in conditional access list.
ACL2NAME	057	80	00	00000008	00	Bin	1 indicator byte; 7 bytes reserved for IBM's use.
ACL2UID	058	80	00	00000008	00	Char	User ID or group.
ACL2ACC	059	80	00	00000001	00	Bin	Access authority.
ACL2ACNT	060	80	00	00000002	00	Int	Access count.

Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	Field being described
ACL2RSVD	061	80	00	00000000	00	Bin	Conditional data. Reserved for IBM's use.
RACLHDR	062	00	00	00000020	00	Bin	RACGLIST header.
RACLDSP	063	00	00	00000000	00	Bin	RACGLIST dataspace information.
FILTERCT	064	10	00	00000004	00		Number of names that Hash to this DIGTNMAP Profile.
FLTRLABL	065	80	00	00000000	00		Label associated with this DIGTNMAP Mapping (matches NMAPLABL for user named by FLTRUSER or user irrmulti.)
FLTRSTAT	066	A0	00	00000001	00		Trust status – bit 0 on for trusted.
FLTRUSER	067	80	00	00000000	00		User ID or criteria profile name.
FLTRNAME	068	80	00	00000000	00		Unhashed issuer's name filter used to create this profile name, (max of 255), followed by a separator, (X'4A'), and the unhashed subject's name filter used to create this profile name (max of 255).
FLTRSVD1	069	80	00	00000000	00		Reserved for IBM's use.
FLTRSVD2	070	80	00	00000000	00		Reserved for IBM's use.
FLTRSVD3	071	80	00	00000000	00		Reserved for IBM's use.
FLTRSVD4	072	80	00	00000000	00		Reserved for IBM's use.
FLTRSVD5	073	80	00	00000000	00		Reserved for IBM's use.
RACDHDR	074	00	08	00000000	00	Bin	CACHECLS header.
DIDCT	075	10	00	00000004	00		Number of names that correspond to this IDIDMAP Profile.
DIDLABL	076	80	00	00000000	00		Label associated with this IDIDMAP class profile mapping (matches DMAPLABL for user named by DIDUSER).
DIDUSER	077	80	00	00000008	00		User ID.
DIDRNAME	078	80	00	00000000	00		Registry name (max of 255).
DIDRSVD1	079	80	00	00000000	00		Reserved for IBM's use.
DIDRSVD2	080	80	00	00000000	00		Reserved for IBM's use.

Field name	Field ID	Flag 1	Flag 2	Combination field IDs					Type	
The following is the COMBINATION segment of the GENERAL template.										
CREADATE	000	40	00	005	000	000	000	000		Combination.
AUTHDATE	000	40	00	005	000	000	000	000		Fields.
AUTHOR	000	40	00	006	000	000	000	000		
TVTOC	000	48	00	025	026	027	028	029		
	000	40	00	030	000	000	000	000		
LOGINFO	000	40	00	032	033	034	000	000		
FIELD	000	40	00	039	040	041	000	000		
ACL	000	40	00	048	049	050	000	000		
ACL1	000	40	00	048	049	000	000	000		
USERDATA	000	40	00	052	053	054	000	000		

General template

Field name	Field ID	Flag 1	Flag 2	Combination field IDs					Type	
ACL2	000	40	00	057	058	059	060	061		Conditional access list
ACL2A3	000	40	00	057	058	059	060	000		Conditional access list
FLTRLST1	000	40	00	065	066	067	068	000		Combo field for FILTER
FLTRLST2	000	40	00	065	067	068	000	000		Combo field for FILTER
CERTRNG	000	40	00	010	011	009	000	000		Digital certificate data.
CERTRNG2	000	40	00	009	011	000	000	000		
CERTRNG3	000	40	00	009	012	013	000	000		
DIDLIST1	000	40	00	076	077	078	000	000		Combination for distributed identity.

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
The following is the SESSION segment of the GENERAL template.							
SESSION	001	00	00	00000000	00		Start of segment fields
SESSKEY	002	00	00	00000000	00	Bin	Session key; maximum length = 8
SLSFLAGS	003	20	00	00000001	00	Bin	Session flag byte Bit Meaning when set 0 SLSLOCK-This profile is locked out 1–7 Reserved for IBM's use
KEYDATE	004	00	00	00000004	00	Date	Last date session key was changed. It is in the format <i>0cyyddF</i> where <i>c</i> =0 for 1900–1999 and <i>c</i> =1 for 2000–2099. For more information on this MVS-returned format, see <i>z/OS MVS Programming: Assembler Services Guide</i> .
KEYINTVL	005	00	00	00000002	00	Int	Number of days before session key expires
SLSFAIL	006	00	00	00000002	00	Int	Current number of invalid attempts
MAXFAIL	007	00	00	00000002	00	Int	Number of invalid attempts before lockout
SENTCNT	008	10	00	00000004	00	Int	Number of session entities in list
SENTITY	009	80	00	00000035	00	Char	Entity name
SENTFLCT	010	80	00	00000002	00	Int	Number of failed attempts for this entity
CONVSEC	011	20	00	00000001	00	Bin	Conversation security. Value Meaning X'40' Conversation security X'50' Persistent verification X'60' User ID and password already verified X'70' User ID and password already verified plus persistent verification X'80' Security none

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
The following is the DLFDATA segment of the GENERAL template.							
DLFDATA	001	00	00	00000000	00		Start of segment fields
RETAIN	002	20	00	00000001	00	Bin	Retain flag byte
JOBNCNT	003	10	00	00000004	00	Int	Count of jobnames
JOBNAME	004	80	00	00000000	00	Char	Jobnames; maximum length = 8
The following is the SSIGNON segment of the GENERAL template.							
SSIGNON	001	00	00	00000000	00		Start of segment fields
SSKEY	002	00	00	00000000	00	Bin	Secured signon key
PTKEYLAB	003	00	00	00000000	00	Char	EPT key label
PTTYPE	004	00	00	00000000	00	Char	PassTicket Type
PTTIMEO	005	00	00	00000004	00	Int	PassTicket Timeout
PTREPLAY	006	00	00	00000001	00	Bin	PassTicket Replay
The following is the STDATA segment of the GENERAL template.							
STDATA	001	00	00	00000000	00		Start of segment fields
STUSER	002	00	00	00000008	40	Char	User ID or =MEMBER
STGROUP	003	00	00	00000008	40	Char	Group name or =MEMBER
FLAGTRUS	004	20	00	00000001	00	Bin	Trusted flag, X'80' = trusted
FLAGPRIV	005	20	00	00000001	00	Bin	Privileged flag, X'80' = privileged
FLAGTRAC	006	20	00	00000001	00	Bin	Trace usage flag X'80' = issue IRR8I2I
The following is the SVFMR segment of the GENERAL template.							
SVFMR	001	00	00	00000000	00		Start of segment fields
SCRIPTN	002	00	00	00000008	00	Char	Script name
PARMN	003	00	00	00000008	00	Char	Parameter name
The following is the CERTDATA segment of the GENERAL template.							
CERTDATA	001	00	00	00000000	00		Start of segment fields
CERT	002	00	00	00000000	00	Bin	Digital certificate
CERTPRVK	003	00	00	00000000	00	Bin	Private key or key label
RINGCT	004	10	00	00000004	00	Int	Number of key rings associated with this certificate
RINGNAME	005	80	00	00000000	00	Char	Profile name of a ring with which this certificate is associated
CERTSTRT	006	00	00	00000000	00		Date and time from which the certificate is valid. If the year is 2041 or earlier, this is an 8-byte TOD format field. If the year is later than 2041, this is the first 8 bytes of an ETOD format field. If the first byte is greater than X'38', the date is in TOD format; otherwise it is in ETOD format.
CERTEND	007	00	00	00000000	00		Date and time after which the certificate is not valid. If the year is 2041 or earlier, this is an 8-byte TOD format field. If the year is later than 2041, this is the first 8 bytes of an ETOD format field. If the first byte is greater than X'38', the date is in TOD format; otherwise it is in ETOD format.

General template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
CERTCT	008	10	00	00000004	00	Int	The number of certificates associated with this key ring. CERTCT is a repeat group that identifies the certificates associated with a key ring. CERTCT is used <i>only</i> with DIGTRING profiles.
CERTNAME	009	80	00	00000000	00	Char	The profile name of the certificate
CERTUSAG	010	80	00	00000004	00	Bin	Certificate usage in ring: <ul style="list-style-type: none"> • X'00000000' – PERSONAL • X'00000001' – SITE • X'00000002' – CERTAUTH
CERTDFLT	011	80	00	00000001	00	Bin	Verifies if it is the default certificate: <ul style="list-style-type: none"> • X'00' – Not the default • X'80' – The default
CERTSJDN	012	80	00	00000000	00	Bin	The subject name of the entity to whom the certificate is issued. This field is a BER-encoded format of the subject's distinguished name as contained in the certificate
CERTLABL	013	80	00	00000000	00	Char	Label associated with the certificate
CERTRSV1	014	80	00	00000000	00		Reserved for IBM's use.
CERTRSV2	015	80	00	00000000	00		Reserved for IBM's use.
CERTRSV3	016	80	00	00000000	00		Reserved for IBM's use.
CERTRSV4	017	80	00	00000000	00		Reserved for IBM's use.
CERTRSV5	018	80	00	00000000	00		Reserved for IBM's use.
CERTRSV6	019	80	00	00000000	00		Reserved for IBM's use.
CERTRSV7	020	80	00	00000000	00		Reserved for IBM's use.
CERTRSV8	021	80	00	00000000	00		Reserved for IBM's use.
CERTRSV9	022	80	00	00000000	00		Reserved for IBM's use.
CERTRSPA	023	80	00	00000000	00		Reserved for IBM's use.
CERTRSPB	024	80	00	00000000	00		Reserved for IBM's use.
CERTRSPC	025	80	00	00000000	00		Reserved for IBM's use.
CERTRSPD	026	80	00	00000000	00		Reserved for IBM's use.
CERTRSPE	027	80	00	00000000	00		Reserved for IBM's use.
CERTRSPF	028	80	00	00000000	00		Reserved for IBM's use.
CERTRSPG	029	80	00	00000000	00		Reserved for IBM's use.
CERTRSPH	030	80	00	00000000	00		Reserved for IBM's use.
CERTRSPI	031	80	00	00000000	00		Reserved for IBM's use.
CERTRSPJ	032	80	00	00000000	00		Reserved for IBM's use.
CERTRSPK	033	80	00	00000000	00		Reserved for IBM's use.

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
CERTPRVT	034	00	00	00000004	00	Bin	Associated key type: <ul style="list-style-type: none"> • X'00000000' – No associated key • X'00000001' – PKCS DER-encoded • X'00000002' – ICSF token label • X'00000003' – PCICC label • X'00000004' – DSA • X'00000005' – ICSF public token label • X'00000006' – Reserved for IBM's use • X'00000007' – NIST ECC key • X'00000008' – Brainpool ECC key • X'00000009' – NIST ECC token label in PKDS • X'0000000A' – Brainpool ECC token label in PKDS • X'0000000B' – RSA token label in TKDS • X'0000000C' – NIST ECC token label in TKDS • X'0000000D' – Brainpool ECC token label in TKDS
CERTPRVS	035	00	00	00000004	00	Int	Private key size in bits
CERTLSER	036	00	00	00000008	00	Bin	The low order 8 bytes of the last certificate that was signed with this key. This field is used with DIGTCERT profiles only
RINGSEQN	037	00	00	00000004	00	Int	Ring change count
CERTGREQ	038	00	00	00000001	00	Bin	Indicates if the certificate is used for generating a request
The following is the TME segment of the GENERAL template.							
TME	001	00	00	00000000	00		Start of segment fields
PARENT	002	00	00	00000000	00	Char	Parent name
CHILDN	003	10	00	00000004	00	Int	Count of children
CHILDREN	004	80	00	00000000	00	Char	Child names
RESN	005	10	00	00000004	00	Int	Count of resource-access specifications
RESOURCE	006	80	00	00000000	00		Resource-access specifications
GROUPN	007	10	00	00000004	00	Int	Count of groups
GROUPS	008	80	00	00000008	00		Group names
ROLEN	009	10	00	00000004	00	Int	Count of role-access specifications
ROLES	010	80	00	00000000	00	Char	Role-access specifications
The following is the KERB segment of the GENERAL template.							
KERB	001	00	00	00000000	00		Start of segment fields
KERBNAME	002	00	00	00000000	00	Char	Kerberos realm name
MINTKTLF	003	00	00	00000000	00	Char	Minimum ticket life
MAXTKTLF	004	00	00	00000000	00	Char	Maximum ticket life
DEFTKTLF	005	00	00	00000000	00	Char	Default ticket life
SALT	006	00	00	00000000	00	Char	Current key salt
ENCTYPE	007	00	00	00000000	00	Char	Encryption type

General template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
CURKEYV	008	00	00	00000000	00	Char	Current key version
CURKEY	009	00	00	00000000	00	Char	Current key value
PREVKEYV	010	00	00	00000000	00	Char	Previous key version
PREVKEY	011	00	00	00000000	00	Char	Previous key value
ENCRYPT	012	00	00	00000004	55	Char	Encryption type
CHKADDRS	013	00	00	00000001	00	Char	Check addresses flag
The following is the PROXY segment of the GENERAL template.							
PROXY	001	00	00	00000000	00		Start of segment fields
LDAPHOST	002	00	00	00000000	00	Char	LDAP server URL; maximum length: 1023
BINDDN	003	00	00	00000000	00	Char	Bind distinguished name; maximum length: 1023
BINDPW	004	00	08	00000000	00	Char	Bind password; maximum length: 128
BINDPWKY	005	00	08	00000071	00	Char	Bind password mask or encrypt key
The following is the EIM segment of the GENERAL template.							
EIM	001	00	00	00000000	00		Start of segment fields
DOMAINDN	002	00	00	00000000	00	Char	EIM Domain Distinguished Names
OPTIONS	003	00	00	00000004	55	Char	EIM Options
LOCALREG	004	00	00	00000000	00	Char	Local Registry Name
KERBREG	005	00	00	00000000	00	Char	Kerberos Registry Name
X509REG	006	00	00	00000000	00	Char	X509 Registry Name
The following is the ALIAS segment of the GENERAL template.							
ALIAS	001	00	00	00000000	00		Start of segment fields
IPLOOK	002	00	10	00000016	00	Bin	IP lookup value
The following is the CDTINFO segment of the GENERAL template.							
CDTINFO	001	00	00	0	0		Start of segment fields
CDTPOSIT	002	00	00	4	FF	Int	POSIT number for class
CDTMAXLN	003	00	00	1	8	Int	Maximum length of profile names
CDTMAXLX	004	00	00	4	FF	Int	Maximum resource or profile name length when using ENTITYX
CDTDFTRC	005	00	00	1	4	Int	Default return code
CDTKEYQL	006	00	00	4	0	Int	Number of key qualifiers
CDTGROUP	007	00	00	8	0	Char	Resource grouping class name
CDTMEMBR	008	00	00	8	0	Char	Member class name

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
CDTFIRST	009	00	00	1	X'C0'	Bin	Character restriction for first character of profile name Value Meaning X'80' Alphabetic X'40' National X'20' Numeric X'10' Special
CDTOTHER	010	00	00	1	X'C0'	Bin	Character restriction for characters of the profile name other than the first character Value Meaning X'80' Alphabetic X'40' National X'20' Numeric X'10' Special
CDTOPER	011	00	00	1	X'00'	Bin	Operations attribute considered Value Meaning X'80' RACF considers OPERATIONS attribute
CDTUACC	012	00	00	1	X'01'	Bin	Default UACC Value Meaning X'80' ALTER X'40' CONTROL X'20' UPDATE X'10' READ X'08' EXECUTE X'04' UACC from ACEE X'01' NONE

General template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
CDTRACL	013	00	00	1	X'00'	Bin	SETROPTS RACLIST Value Meaning X'00' RACLIST disallowed X'80' RACLIST allowed X'40' RACLIST required
CDTGENL	014	00	00	1	X'00'	Bin	SETROPTS GENLIST Value Meaning X'80' GENLIST allowed
CDTPRFAL	015	00	00	1	X'80'	Bin	Profiles allowed Value Meaning X'80' Profiles are allowed
CDTSLREQ	016	00	00	1	X'00'	Bin	Security labels required Value Meaning X'80' Security labels are required
CDTMAC	017	00	00	1	X'80'	Bin	Mandatory access checking (MAC) processing Value Meaning X'80' Normal mandatory access checks X'40' Reverse mandatory access checks X'20' Equal mandatory access checks
CDTSIGL	018	00	00	1	X'00'	Bin	ENF Signal Value Meaning X'80' ENF signal to be sent
CDTCASE	019	00	00	1	X'00'	Bin	Case of profile names Value Meaning X'00' Uppercase X'80' ASIS - preserve case

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
CDTGEN	020	00	00	1	X'80'	Bin	SETROPTS GENERIC Value Meaning X'80' GENERIC allowed
The following is the ICTX segment of the GENERAL template.							
ICTX	001	00	00	00000000	00		Start of segment fields
USEMAP	002	00	00	00000001	80	Bin	Application supplied mapping Value Meaning X'80' Use the mapping
DOMAP	003	00	00	00000001	00	Bin	Identity cache mapping Value Meaning X'80' Do the mapping
MAPREQ	004	00	00	00000001	00	Bin	Value Meaning X'80' Mapping is required
MAPTIMEO	005	00	00	00000002	00	Int	Mapping timeout adjustment
The following is the CFDEF segment of the GENERAL template.							
CFDEF	001	00	00	0	0		Start of segment fields for defining custom field attributes
CFDTYPE	002	00	00	1	01	Bin	Data type for custom field: <ul style="list-style-type: none"> • 01 - character • 02 - numeric • 03 - flag • 04 - hex
CFMXLEN	003	00	00	4	FF	Int	Maximum field length
CFMXVAL	004	00	00	4	FF	Int	Maximum numeric value
CFMIVAL	005	00	00	4	FF	Int	Minimum numeric value
CFFIRST	006	00	00	1	00	Bin	First character restrictions: <ul style="list-style-type: none"> • 01 - alpha • 02 - alphanum • 03 - any • 04 - nonatabc • 05 - nonatnum • 06 - numeric

General template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
CFOTHER	007	00	00	1	00	Bin	Other character restrictions: <ul style="list-style-type: none"> • 01 - alpha • 02 - alphanum • 03 - any • 04 - nonatabc • 05 - nonatnum • 06 - numeric
CFMIXED	008	20	00	1	00	Bin	If bit 0 is on, mixed case is allowed
CFHELP	009	00	00	00	00	Char	Help text; maximum length = 255
CFLIST	010	00	00	00	00	Char	List heading text; maximum length = 40
CFVALRX	011	00	00	00	00	Char	Custom field REXX validation exit
CFACEE	012	20	00	1	00	Bin	ACEE(YES/NO)
The following is the SIGVER segment of the GENERAL template.							
SIGVER	001	00	00	0	0		Start of segment fields
SIGREQD	002	00	00	1	0	Bin	Module must have a signature: Value Meaning X'80' Yes X'00' No
FAILLOAD	003	00	00	1	0	Bin	Loader failure conditions: Value Meaning X'80' Bad signature only X'40' Any failing signature condition X'00' Never
SIGAUDIT	004	00	00	1	0	Bin	RACF audit conditions: Value Meaning X'80' Bad signature only X'40' Any failing signature condition X'20' Success X'01' All X'00' None
The following is the ICSF segment of the GENERAL template.							
ICSF	01	00	00	00000000	00		Start of segment fields for defining ICSF attributes

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
CSFSEXP	02	00	00	00000001	00	Bin	Symmetric key export option: Value Meaning X'80' BYLIST X'40' BYNONE X'00' BYANY
CSFSKLCT	03	10	00	00000004	00	Int	Count of PKDS labels
CSFSKLBS	04	80	00	00000000	00	Char	PKDS labels that might be used to export this symmetric key
CSFSCLCT	05	10	00	00000004	0	Int	Count of certificate labels
CSFSCLBS	06	80	00	00000000	00	Char	Certificate labels that might be used to export this symmetric key
CSFAUSE	07	00	00	00000004	55	Bin	Asymmetric key usage. In byte 3: Value Meaning X'08' NOSECUREEXPORT X'04' SECUREEXPORT X'02' NOHANDSHAKE X'01' HANDSHAKE
CSFSCPW	08	00	00	00000001	00	Bin	Symmetric key CPACF wrap Value Meaning X'80' YES X'00' NO
CSFSCPR	09	00	00	00000001	00	Bin	Symmetric key CPACF return Value Meaning X'80' YES X'00' NO
The following is the MFA segment of the GENERAL template.							
MFA	001	00	00	00000000	00		Start of segment fields
MFDATA	002	00	00	00000000	00		Free-form factor metadata
The following is the MFPOLICY segment of the GENERAL template.							
MFPOLICY	001	00	00	00000000	00		Start of segment fields
MFFCTRN	002	10	00	00000004	00		Number of factors in policy
MFFCTRS	003	80	00	00000000	00		Policy factor list

Reserved template

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
MFTIMEO	004	00	00	00000004	00		Policy token timeout
MFREUSE	005	00	00	00000001	00		Policy reuse setting
The following is the CSDATA segment of the GENERAL template.							
CSDATA	001	00	00	00000000	00	Bin	Start of the segment fields for custom fields. Note: Intended usage for these fields is dictated by your installation. See <i>z/OS Security Server RACF Security Administrator's Guide</i> for more information on custom fields.
CSCNT	002	10	00	00000004	00	Char	Count of custom fields
CSTYPE	003	80	00	00000001	01	Char	Custom field type <ul style="list-style-type: none"> • 01 - character • 02 - numeric • 03 - flag • 04 - hex
CSKEY	004	80	00	00000000	00	Char	Custom field keyword
CSVALUE	005	80	00	00000000	00	Char	Custom field value
The following is the IDTPARMS segment of the GENERAL template.							
IDTPARMS	001	00	00	00000000	00		Start of segment field
IDTTOKN	002	00	00	00000000	00	Char	PKCS#11 Token Name
IDTSEQN	003	00	00	00000000	00	Char	PKCS#11 Sequence Number
IDTCAT	004	00	00	00000000	00	Char	PKCS#11 Category
IDTSALG	005	00	00	00000000	00	Char	Signature Algorithm
IDTTIMEO	006	00	00	00000004	00	Int	IDT Timeout
IDTANYAP	007	00	00	00000001	80	Bin	IDT Any Application
IDTPROTA	008	00	00	00000001	80	Bin	IDT Protected allowed
IDTLABP	009	00	00	00000000	00	Char	Primary Label
IDTKIDP	010	00	00	00000000	00	Char	Primary KID
IDTLABB	011	00	00	00000000	00	Char	Reserved
IDTKIDB	012	00	00	00000000	00	Char	Reserved

Field name	Field ID	Flag 1	Flag 2	Combination field IDs					Type	
The following is a COMBINATION field of the CSDATA segment of the GENERAL template.										
CSCDATA	000	40	00	003	004	005	000	000	Char	Combination field for custom fields

Reserved template for the RACF database

This template is reserved for IBM's use. The installation must not use it.

The contents of the reserved template are as follows:

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
RSVTMP03	001	00	00	00000000	00		
ENTYPE	002	00	00	00000001	00		The number corresponding to the type of profile being described.
VERSION	003	00	00	00000001	00		Template version number.

Appendix E. Event code qualifier descriptions

Event codes and event code qualifiers

The RACF event code (found in the SMF80EVT field of the SMF record) and the RACF event code qualifier (found in the SMF80EVQ field of the SMF record) are determined during RACF processing. This topic explains the meaning of each qualifier code by event.

Event 1(1): JOB INITIATION/TSO LOGON/TSO LOGOFF

This event is logged by RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX. Installation exit ICHRIX02 can change the return code of the RACROUTE REQUEST=VERIFY or RACROUTE REQUEST=VERIFYX request to any value. The return code significantly influences the corresponding audit record's event code 1 qualifier. You should be familiar with any ICHRIX02 processing in effect for your installation. See *z/OS Security Server RACF System Programmer's Guide* for details.

For this event, code qualifiers 0 and 8 do not exist as type 80 records. They are contained in the unloaded records from the RACF SMF data unload utility (IRRADU00) and as reports and reformatted records from the RACF report writer (RACFRW).

The explanations of the event code qualifiers for Event 1 are:

0(0)

SUCCESSFUL INITIATION The job began successfully.

1(1)

INVALID PASSWORD The password specified on the job card or at logon is incorrect.

2(2)

INVALID GROUP The user tried to log on or to initiate a job using a group that the user is not a member of.

3(3)

INVALID OIDCARD Operator identification cards are used at the installation, and the data received from the one used does not match that of the user's profile.

4(4)

INVALID TERMINAL/CONSOLE The user is not authorized to the port of entry (POE). There are four kinds of POEs, each with its own profile class: APPCPORT, CONSOLE, JESINPUT, and TERMINAL. One of the following occurred:

- The port of entry is active but the user is not authorized.
- The user is denied access because of conditional days/times in the user profile.
- The user is denied access because of conditional days/times in the class profile (TERMINAL class only).

5(5)

INVALID APPLICATION The APPL class is active, and the user is trying to log on to an application without authorization.

6(6)

REVOKED USER ID ATTEMPTING ACCESS The user ID specified on the logon or job card has been revoked. One of the following occurred:

- The installation-defined limit of password attempts was reached at an earlier time.
- The inactive interval was reached.
- The revoke date in the user's profile is in effect.
- The RACF administrator revoked the user ID.

The RACF administrator must reset the user ID before the user can log on again.

7(7)

USER ID AUTOMATICALLY REVOKED The user ID has been automatically revoked. The installation-defined limit of password and password phrase attempts was reached.

8(8)

SUCCESSFUL TERMINATION The job completed successfully.

9(9)

UNDEFINED USER ID The user ID specified on the job card or at logon is not defined to the RACF database.

10(A)

INSUFFICIENT SECURITY LABEL AUTHORITY One of the following occurred:

- SETROPTS MLS FAILURES is in effect and the user's security label does not dominate the submitter's security label. Two exceptions are explained under Qualifier 20.
- SETROPTS MLACTIVE FAILURES is in effect and the job card/logon attempt does not specify a valid security label. One exception is explained under Qualifier 21.

11(B)

NOT AUTHORIZED TO SECURITY LABEL The user is not authorized to the security label specified. One exception is explained under Qualifier 22.

12(C)

SUCCESSFUL RACINIT INITIATION The job or user was verified.

13(D)

SUCCESSFUL RACINIT DELETE The job completed or the user logged off.

14(E)

SYSTEM NOW REQUIRES MORE AUTHORITY SETROPTS MLQUIET is in effect. If this is a user verification, the user is not a console operator and does not have the SPECIAL attribute. If this is a job verification, the job is not part of the trusted computing base (TCB). The verification fails.

15(F)

REMOTE JOB ENTRY—JOB NOT AUTHORIZED The submitting node is not authorized to the system; a NODES profile prevents remote job entry. The profile has the format 'submit_node.RUSER.userid' and has a UACC of NONE.

Note:

Surrogate Function Qualifiers:

Qualifiers 16, 17, and 18 involve the use of the surrogate function, and occur if any of the following conditions is met:

- The SURROGAT class is active.
- General resource profiles of the SURROGAT class are defined for the job card's user ID, and the user ID submitting the job is permitted to the profile with at least READ access.
- The submitter is authorized to the security label of the job.

For more information, see [*z/OS Security Server RACF Security Administrator's Guide*](#).

16(10)

SURROGATE CLASS IS INACTIVE The SURROGAT class is inactive. The job card has a user ID that is different from the submitter's user ID, and there is no password specified.

17(11)

SUBMITTER IS NOT AUTHORIZED BY USER The SURROGAT class is active. Either there is no SURROGAT profile for the job card's user ID, or the submitter's user ID is not permitted to the profile.

18(12)

SUBMITTER IS NOT AUTHORIZED TO SECURITY LABEL The SECLABEL class is active and there is a security label on the job card. The submitter is not authorized to the security label specified on the job card.

19(13)

USER IS NOT AUTHORIZED TO JOB The JESJOBS class is active, and the user is not authorized to the jobname.

20(14)

WARNING—INSUFFICIENT SECURITY LABEL AUTHORITY One of the following occurred:

- SETROPTS MLS WARNING is in effect and the security label on the job card does not dominate the submitter's security label.
- SETROPTS MLS FAILURES is in effect, the user's security label does not dominate the submitter's, and the user has the SPECIAL attribute.
- SETROPTS MLS FAILURES and SETROPTS COMPATMODE are in effect, the user's security label does not dominate the submitter's, and the submitter's or the job owner's security label is the default.

The verification does not fail.

21(15)

WARNING—SECURITY LABEL MISSING FROM JOB, USER, OR PROFILE One of the following occurred:

- MLACTIVE WARNING is in effect, and the job card or logon attempt did not specify a valid security label.
- MLACTIVE FAILURES is in effect, the user has the SPECIAL attribute, and a valid security label is not specified.

The verification does not fail.

22(16)

WARNING—NOT AUTHORIZED TO SECURITY LABEL The user has the SPECIAL attribute, the security label is SYSHIGH, and the user does not have authority to it. The verification does not fail.

23(17)

SECURITY LABELS NOT COMPATIBLE SETROPTS MLS is not active, the submitter's user ID is different from the user ID on the job card, and the submitter's and the user's security labels are disjoint (neither one dominates the other).

One exception is listed under Qualifier 24.

24(18)

WARNING—SECURITY LABELS NOT COMPATIBLE SETROPTS MLS is not active, the submitter's user ID is different from the user ID on the job card, the submitter's and user's security labels are disjoint, SETROPTS COMPATMODE is in effect, and the submitter's or user's security label is the default. The verification does not fail.

25(19)

CURRENT PASSWORD HAS EXPIRED The user's password has expired for one of the following reasons:

- The installation specification in SETROPTS PASSWORD INTERVAL command
- Creation of the password in the ADDUSER command
- Alteration of the password with the ALTUSER PASSWORD command

26(1A)

INVALID NEW PASSWORD The new password specified may be incorrect because:

- It is all blanks.
- The characters are not all alphanumeric.
- The characters do not match the installation's password syntax rules (set by the SETROPTS PASSWORD command).
- It is the same as a past password (the extent of the past history determined by the SETROPTS PASSWORD HISTORY command).
- It is marked invalid by the installation's password exit.

- It is too soon to change the password (as determined by the SETROPTS PASSWORD MINCHANGE command).

27(1B)

VERIFICATION FAILED BY INSTALLATION The installation exit ICHRIX01 or ICHRIX02 failed the request.

28(1C)

GROUP ACCESS HAS BEEN REVOKED The user's membership to the group specified has been revoked.

29(1D)

OIDCARD IS REQUIRED An OIDCARD is required by the installation but none was given.

30(1E)

NETWORK JOB ENTRY—JOB NOT AUTHORIZED For session types of NJE SYSOUT or NJE BATCH, the verification fails because one of the following occurred:

- The user, group, or security label requirements in the NODES profiles were not met.
- The submitter's node is not valid.
- The reverify check failed.

See *z/OS Security Server RACF Security Administrator's Guide* for details on NJE.

31(1F)

WARNING—UNKNOWN USER FROM TRUSTED NODE PROPAGATED The combination of having a trusted node submit a job with the undefined user ID warrants this logging. The verification does not fail.

For an NJE BATCH job, the submitting user is the NJE undefined user ID. The default NJE undefined user ID is eight question marks (????????), unless it was changed with the SETROPTS JES NJEUSERID command. The submitting node is trusted (its best-fit NODES profile on the receiving node's system has a UACC of at least UPDATE). This profile allows propagation of submitters; however, the undefined user ID does not propagate.

32(20)

SUCCESSFUL INITIATION USING PASSTICKET Logon was achieved using a PassTicket.

33(21)

ATTEMPTED REPLAY OF PASSTICKET Logon was rejected because of attempted replay of a PassTicket.

34(22)

CLIENT SECURITY LABEL NOT EQUIVALENT TO SERVER'S Logon was rejected because security labels are not equivalent.

35(23)

USER AUTOMATICALLY REVOKED DUE TO INACTIVITY A user has not logged on, submitted a job or accessed the system for so long that the user ID has become inactive. RACF prevents the user from accessing the system.

36(24)

PASS PHRASE IS NOT VALID A user attempted to access the system specifying a password phrase that is not valid or specifying a password phrase for a protected user ID. RACF prevents the user from accessing the system.

37(25)

NEW PASS PHRASE IS NOT VALID Logon was rejected because the new password phrase is not valid.

38(26)

CURRENT PASS PHRASE HAS EXPIRED Logon was rejected because the current password phrase has expired.

39(27)

NO RACF USER ID FOUND FOR DISTRIBUTED IDENTITY Logon was rejected because no RACF user ID was found for the distributed identity.

40(28)

Successful Multi-Factor Authentication (MFA) Logon was successful for multi-factor authentication.

41(29)

Failed Multi-Factor Authentication (MFA) Logon was rejected due to nonvalid multi-factor authentication credentials.

42(2A)

Failed Authentication Authentication failed because no multi-factor decision could be made for an MFA user who has the NOPWFALLBACK option.

43(2B)

IBM MFA partial success The credentials were not incorrect, but a re-authentication is required.

44(2C)

Identity Token validation error An error was encountered validating a provided identity token.

45(2D)

Identity Token build error An error was encountered attempting to build an identity token.

46(2E)

Failed Identity Token authentication Signature validation of a provided identity token failed.

Event 2(2): RESOURCE ACCESS

This event is logged by RACROUTE REQUEST=AUTH, RACROUTE REQUEST=DIRAUTH and RACROUTE REQUEST=FASTAUTH.

The explanations of the event code qualifiers for Event 2 are:

0(0)

SUCCESSFUL ACCESS The user has authorization to the resource.

1(1)

INSUFFICIENT AUTHORITY The user does not have authorization to the resource.

2(2)

PROFILE NOT FOUND—RACFIND SPECIFIED ON MACRO If the request is AUTH, the RACFIND keyword equaled YES on the authorization request, specifying that a discrete profile should exist for the resource. No discrete or generic RACF protection was found.

If the request is FASTAUTH, the program is not controlled and the PADS data sets are open.

3(3)

ACCESS PERMITTED DUE TO WARNING The user does not have proper authority to the resource. However, the resource's profile has the WARNING option and allows the access.

Note:

Exceptions

- PROGRAM class profiles cannot use the WARNING option.
- RACLISTed profiles use the WARNING option only if they are RACLISTed by SETROPTS or a RACROUTE REQUEST=LIST that specifies RELEASE=1.8 or later.

4(4)

FAILED DUE TO PROTECTALL SETROPTS PROTECTALL FAILURES is in effect, and the data set has not been protected by a discrete or generic profile.

Note:

Exceptions

- A privileged user bypasses this checking (no auditing done).
- A trusted user bypasses the checking, but can be audited with the SETROPTS LOGOPTIONS command.
- A user with the SPECIAL attribute gets a warning (see Qualifier 5).

- A system-generated temporary data set does not require protection.

5(5)

WARNING ISSUED DUE TO PROTECTALL SETROPTS PROTECTALL WARNING is in effect, and the data set has not been protected by a discrete or generic profile. The authorization request does not fail.

The exceptions in Qualifier 4 also apply.

6(6)

INSUFFICIENT CATEGORY/SECLEVEL The installation uses categories or security levels as separate entities. One of the following occurred:

- The user's SECLEVEL is less than the SECLEVEL of the resource.
- The user is not a member of every CATEGORY associated with the resource.

7(7)

INSUFFICIENT SECURITY LABEL AUTHORITY The SECLABEL class is active and one of the following occurred:

- The user's security label does not dominate the resource's.
- The user does not have a security label, but the resource does.
- SETROPTS MACTIVE FAILURES is in effect, and either the user or the resource is missing a security label. One exception is explained in Qualifier 8.
- The resource's class requires reverse domination checking, and the resource's security label does not dominate the user's.
- SETROPTS MLS FAILURES is in effect; the user's security label does not equal the resource's, and the requested access is UPDATE or CONTROL. One exception is explained under Qualifier 9.

8(8)

SECURITY LABEL MISSING FROM JOB, USER OR PROFILE One of the following occurred:

- SETROPTS MACTIVE WARNING is in effect, the SECLABEL class is active, and either the resource or user is missing a security label.
- SETROPTS MACTIVE FAILURES is in effect, the user has the SPECIAL attribute, and either the resource or the user is missing a security label.

9(9)

WARNING—INSUFFICIENT SECURITY LABEL AUTHORITY One of the following occurred:

- The SECLABEL class is active, SETROPTS MLS WARNING is in effect, the user's security label does not equal the resource's security label, and the requested access is UPDATE or CONTROL.
- SETROPTS MLS FAILURES is in effect, the user's security label does not equal the resource's security label, the requested access is UPDATE or CONTROL, and the user has the SPECIAL attribute.

10(A)

WARNING—DATA SET NOT CATALOGED SETROPTS CATDSNS WARNING is in effect. The data set being accessed cannot be cataloged.

See *z/OS Security Server RACF Command Language Reference* for more information.

11(B)

DATA SET NOT CATALOGED SETROPTS CATDSNS FAILURES is in effect. The data set being accessed cannot be cataloged. If the user has the SPECIAL attribute, only a warning is issued (see Qualifier 10).

See *z/OS Security Server RACF Command Language Reference* for more information.

12(C)

PROFILE NOT FOUND—REQUIRED FOR AUTHORITY CHECKING A profile was not found for the general resource, and that resource's class has a default return code greater than 4. The authorization request fails.

13(D)

WARNING—INSUFFICIENT CATEGORY/SECLEVEL The installation uses categories or security levels as separate entities. One of the following occurred:

- The user's SECLEVEL is less than the SECLEVEL of the resource.
- The user is not a member of every CATEGORY associated with the resource.

The resource profile has the WARNING option, so access is given.

Note:

Exceptions

- PROGRAM class profiles cannot use the WARNING option.
- RACLISTed profiles can use the WARNING option only if they are RACLISTed by SETROPTS or a RACF 1.8 (or later) RACROUTE REQUEST=LIST.

14(E)

WARNING—NON-MAIN EXECUTION ENVIRONMENT Non-MAIN execution environment was detected while in ENHANCED PGMSECURITY mode. Conditional access for Program Access to Data Sets (PADS) or access to EXECUTE-controlled program is temporarily allowed.

15(F)

CONDITIONAL ACCESS ALLOWED VIA® BASIC MODE PROGRAM Conditional access for Program Access to Data Sets (PADS) or access to EXECUTE-controlled program is allowed through the BASIC mode program while in ENHANCED PGMSECURITY mode.

Event 3(3): ADDVOL/CHGVOL

This event refers to RACROUTE REQUEST=DEFINE,TYPE=ADDVOL and RACROUTE REQUEST=DEFINE,TYPE=CHGVOL.

The explanations of the event code qualifiers for Event 3 are:

0(0)

SUCCESSFUL PROCESSING OF NEW VOLUME One of the following occurred:

- The user has proper administrative authority to the DATASET profile; in the case of tape data sets with TAPEVOL active, the user also had administrative authority to the TAPEVOL profile.
- SETROPTS MLS WARNING is in effect, the TAPEVOL class is active, a TAPEVOL profile exists, and the user's security label does not equal the resource's.
- SETROPTS MLACTIVE WARNING is in effect, the TAPEVOL class is active, and no TAPEVOL profile exists for the volume.

1(1)

INSUFFICIENT AUTHORITY The user did not have administrative authority to the DATASET profile, or, in the case of tape data sets, the TAPEVOL class is active and the user did not have administrative authority to the TAPEVOL profile.

2(2)

INSUFFICIENT SECURITY LABEL AUTHORITY The SECLABEL class is active, the data set is a tape data set, the TAPEVOL class is active, and the user's security label does not dominate the security label found in the TAPEVOL profile.

3(3)

LESS SPECIFIC PROFILE EXISTS WITH DIFFERENT SECURITY LABEL The SECLABEL class is active, SETROPTS MLSTABLE is in effect, a less specific generic profile exists that does not have the same security label, the data set is a tape data set, and the TAPEVOL class is active. Changing the volume would change the TAPEVOL profile's security label, violating SETROPTS MLSTABLE rules.

Note:

Exception

If SETROPTS MLQUIET is also in effect and the user has the SPECIAL attribute, the request does not fail and this event is not logged.

Event 4(4): RENAME RESOURCE

This event is based on RACROUTE REQUEST=DEFINE,TYPE=DEFINE,NEWNAME or RACROUTE REQUEST=DEFINE,TYPE=DEFINE,NEWNAMX.

The explanations of the event code qualifiers for Event 4 are:

0(0)

SUCCESSFUL RENAME One of the following occurred:

- The user has sufficient authority to rename the resource.
- The SECLABEL class is active, SETROPTS MLACTIVE WARNING is in effect, and the user or the resource does not have a security label.

1(1)

INVALID GROUP The resource to be renamed is a data set, and the high-level qualifier of the new data set is not a valid group or user ID.

2(2)

USER NOT IN GROUP The resource is a data set, RACFIND is not set to NO, the high-level qualifier of the new data set name is a group, and the user does not belong to that group.

3(3)

INSUFFICIENT AUTHORITY One of the following occurred:

- SETROPTS GENERICOWNER is in effect, and renaming the profile would violate GENERICOWNER rules.
- The resource is a data set, and the high-level qualifier is a group or user ID. The user is not authorized to create a new data set by the generic profile protecting the new name, and the high-level qualifier of the new data set name is beyond the scope of the user.
- The resource is an SFS file or directory, and the second qualifier is a user ID. The user is not authorized to create a new file or directory by the generic profile protecting the new name, and the second qualifier of the new file or directory name is beyond the scope of the user.

See *z/OS Security Server RACF Security Administrator's Guide*.

4(4)

RESOURCE NAME ALREADY DEFINED The requested new name already has a discrete profile defined. The return code of the RENAME is 4.

5(5)

USER NOT DEFINED TO RACF The installation's naming convention routine has indicated that the high-level qualifier is a user ID that is not defined to RACF. One of the following occurred:

- RACFIND is not set to NO.
- The resource is protected by a generic or global profile, and the user does not have ALTER access to it.

6(6)

RESOURCE NOT PROTECTED SETROPTS PROTECTALL FAILURES is in effect, and the new data set name is not protected by a profile.

7(7)

WARNING—RESOURCE NOT PROTECTED SETROPTS PROTECTALL WARNINGS is in effect, and the new data set name is not protected by a profile.

The RENAME is allowed.

8(8)

USER IN SECOND QUALIFIER IS NOT RACF DEFINED The second qualifier of the new name is not a valid user ID.

9(9)

LESS SPECIFIC PROFILE EXISTS WITH DIFFERENT SECURITY LABEL The SECLABEL class is active, SETROPTS MLSTABLE is in effect, and there is a less specific generic profile existing for the new name with a different security label. Renaming this resource would violate SETROPTS MLSTABLE rules.

10(A)

INSUFFICIENT SECURITY LABEL AUTHORITY The SECLABEL class is active, SETROPTS MLS FAILURES is in effect, and the user is not authorized to the security label of the resource to be renamed.

11(B)

RESOURCE NOT PROTECTED BY SECURITY LABEL The SECLABEL class is active, SETROPTS MLS FAILURES is in effect, and the profile covering the old resource name does not have a security label.

12(C)

NEW NAME NOT PROTECTED BY SECURITY LABEL The SECLABEL class is active, SETROPTS MLS FAILURES is in effect, and the profile that would cover the new resource name does not have a security label.

13(D)

NEW SECURITY LABEL MUST DOMINATE OLD SECURITY LABEL The SECLABEL class is active, SETROPTS MLS FAILURES is in effect, and the security label of the profile covering the new resource name does not dominate the security label of the profile covering the old resource name.

14(E)

INSUFFICIENT SECURITY LABEL AUTHORITY The SECLABEL class is active, SETROPTS MLS WARNING is in effect, and the user is not authorized to the security label of the profile. The RENAME is allowed.

15(F)

WARNING—RESOURCE NOT PROTECTED BY SECURITY LABEL The SECLABEL class is active, SETROPTS MLS WARNING is in effect, and the profile covering the old resource name does not have a security label. The RENAME is allowed.

16(10)

WARNING—NEW NAME NOT PROTECTED BY SECURITY LABEL The SECLABEL class is active, SETROPTS MLS WARNING is in effect, and the profile that would cover the new resource name does not have a security label. The RENAME is allowed.

17(11)

WARNING—NEW SECURITY LABEL MUST DOMINATE OLD SECURITY LABEL The SECLABEL class is active, SETROPTS MLS WARNING is in effect, and the security label of the profile covering the new resource name does not dominate the security label of the profile covering the old resource name. The RENAME does not fail.

Event 5(5): DELETE RESOURCE

This event is based on RACROUTE REQUEST=DEFINE,TYPE=DELETE.

The explanations of the event code qualifiers for Event 5 are:

0(0)

SUCCESSFUL SCRATCH The resource profile was deleted.

1(1)

RESOURCE NOT FOUND The resource profile was not found.

2(2)

INVALID VOLUME The class is DATASET, and the data set does not reside on the volume specified.

Event 6(6): DELETE ONE VOLUME OF A MULTIVOLUME RESOURCE

This event is based on RACROUTE REQUEST=DEFINE,TYPE=DELETE.

The explanations of the event code qualifiers for Event 6 are:

0(0)

SUCCESSFUL DELETION The volume was successfully deleted from the DATASET profile.

Event 7(7): DEFINE RESOURCE

This event is based on RACROUTE REQUEST=DEFINE,TYPE=DEFINE.

The explanations of the event code qualifiers for Event 7 are:

0(0)

SUCCESSFUL DEFINITION

- The user had sufficient authority to define the resource.
- The SECLABEL class is active, SETROPTS MLACTIVE WARNING is in effect, and the user or the resource does not have a security label.

1(1)

GROUP UNDEFINED The resource to be defined is a data set, and the high-level qualifier is not a valid group or user ID.

2(2)

USER NOT IN GROUP The resource is a data set, RACFIND is not set to NO, the high-level qualifier is a group, and the user does not belong to that group.

3(3)

INSUFFICIENT AUTHORITY One of the following occurred:

- SETROPTS GENERICOWNER is in effect and defining the profile would violate GENERICOWNER rules.
- For general resources, the user is not authorized to define profiles in the class.
- The resource is a data set, and the high-level qualifier of the resource is a group or user ID. The user is not authorized to create a new data set by the generic profile protecting the new name, and the high-level qualifier of the new data set name is beyond the scope of the user.
- The resource is an SFS file or directory, and the second qualifier is a user ID. The user is not authorized to create a new file or directory by the generic profile protecting the new name, and the second qualifier of the new file or directory name is beyond the scope of the user.

See *z/OS Security Server RACF Security Administrator's Guide*.

4(4)

RESOURCE NAME ALREADY DEFINED The requested name already has a discrete profile defined. The return code of the DEFINE is 4.

5(5)

USER NOT DEFINED TO RACF The installation's naming convention routine has indicated that the high-level qualifier is a user ID that is not defined to RACF. One of the following occurred:

- RACFIND is not set to NO.
- The resource is protected by a generic or global profile, and the user does not have ALTER access to it.

6(6)

RESOURCE NOT PROTECTED SETROPTS PROTECTALL FAILURES is in effect, and the data set to be defined is not protected by a profile.

7(7)

WARNING—RESOURCE NOT PROTECTED SETROPTS PROTECTALL WARNINGS is in effect, and the data set to be defined is not protected by a profile. The DEFINE is allowed.

8(8)

WARNING—SECURITY LABEL MISSING FROM JOB, USER, OR PROFILE The SECLABEL and TAPEVOL classes are active. SETROPTS MLACTIVE WARNING is in effect, and the TAPEVOL profile is without a security label. The DEFINE is allowed.

9(9)

INSUFFICIENT SECURITY LABEL AUTHORITY The SECLABEL and TAPEVOL classes are active. SETROPTS MLS WARNING is in effect, and the user's security label does not dominate the one found in the TAPEVOL profile.

The DEFINE is allowed.

10(A)

USER IN SECOND QUALIFIER IS NOT RACF-DEFINED The second qualifier of the name is not a valid user ID.

11(B)

INSUFFICIENT SECURITY LABEL AUTHORITY The SECLABEL class is active, and one of the following occurred:

- SETROPTS MLACTIVE FAILURES is in effect, and the user is missing a security label.
- SETROPTS MLACTIVE FAILURES is in effect, and the resource is missing a security label.
- The user's security label does not dominate the resource's.
- SETROPTS MLS FAILURES is in effect, and the user's security label does not equal the resource's.

12(C)

LESS SPECIFIC PROFILE EXISTS WITH A DIFFERENT SECURITY LABEL The SECLABEL class is active, SETROPTS MLSTABLE is in effect, and there is a less specific generic profile existing for the name with a different security label.

Defining this resource would violate SETROPTS MLSTABLE rules.

Event 8(8)–25(19): COMMANDS

Events 8 through 25 apply to the RACF commands. The following qualifier codes are used for each event:

0(0)

NO VIOLATIONS DETECTED The RACF command was issued successfully. This qualifier applies to all RACF commands.

1(1)

INSUFFICIENT AUTHORITY The user did not have the authority to issue the RACF command. This qualifier applies to all RACF commands.

2(2)

KEYWORD VIOLATIONS DETECTED The user had the authority to issue the RACF command, but not to all the keywords that were specified. Keywords that the user is not authorized to use are ignored. For example, a user with the SPECIAL attribute but without the AUDITOR attribute can issue the ALTUSER command, but not with the GLOBALAUDIT keyword. This qualifier applies to all RACF commands.

3(3)

SUCCESSFUL LISTING OF DATASETS This logs the successful use of LISTDSD DSNS.

4(4)

SYSTEM ERROR IN LISTING OF DATA SETS This logs an error in attempting LISTDSD DSNS.

Note:

1. When the SETROPTS command is issued with a keyword that contains an asterisk (*), the asterisk is displayed in the output. For example, if you issue the command SETROPTS AUDIT(*), the output contains AUDIT(*)
2. When the SETROPTS command is issued with a keyword that lists more than ten classes, the output lists the first ten classes and displays the remaining number as an ellipsis. For example, if you issue the command SETROPTS CLASSACT(class1 class2 class3 class4 class5 class6 class7 class8 class9 class10 class11 class12), the output appears as CLASSACT(class1 class2 class3 class4 class5 class6 class7 class8 class9 class10 ...(00002)).

3. When the RVARY command is issued, the DATASET keyword lists the names of as many RACF databases as can fit in the 1024 character output. The remainder are shown as an ellipsis (...nnnnn).
4. When the RVARY command is issued with the NOCLASSACT(*) keyword or with more than ten classes specified, the output lists the first ten classes. The remaining classes are shown as an ellipsis.

Event 26(1A): APPCLU

This event is logged by RACROUTE REQUEST=AUDIT,EVENT='APPCLU'. This event applies to establishing a session between two logical units (referred to as the local LU and the partner LU) in accordance with the System Network Architecture (SNA). VTAM® and CICS call RACF for security information stored in general resource profiles in the APPCLU class.

Each profile contains an 8-byte session key that is used in verification; the two LUs must have corresponding profiles with identical keys so that the handshaking of encrypted data is successful.

The explanations of the event code qualifiers for Event 26 are:

0(0)

PARTNER VERIFICATION WAS SUCCESSFUL The handshaking was successful. The LUs established a connection.

1(1)

SESSION ESTABLISHED WITHOUT VERIFICATION No handshaking was done, but the LUs were still allowed to establish a connection, with the knowledge that the partners were not verified.

2(2)

LOCAL LU KEY WILL EXPIRE IN 5 DAYS OR LESS The handshaking was successful; this qualifier was set to tell users when the local LU's session key would expire.

3(3)

PARTNER LU ACCESS HAS BEEN REVOKED Too many unsuccessful attempts were made at matching the session key.

4(4)

PARTNER LU KEY DOES NOT MATCH THIS LU KEY An attempt was made to establish a session, but the session keys did not match. For example, the two sets of identical data encrypted with the two keys did not match.

5(5)

SESSION TERMINATED FOR SECURITY REASONS One or both of the APPCLU profiles involved have the keyword LOCK specified in their session information, preventing any connections from being made. This keyword enables the security administrator to temporarily prevent specific connections without deleting any profiles.

6(6)

REQUIRED SESSION KEY NOT DEFINED The local LU had VERIFY=REQUIRED coded on its APPL statement, indicating that session level verification must be used on all sessions with the LU. One of the following occurred:

- The local LU is the primary LU and no password was defined in RACF for the LU pair.
- The partner LU is the primary LU, but the bind it sent to the local LU did not contain random data (which would indicate that the partner is using session level verification also).

7(7)

POSSIBLE SECURITY ATTACK BY PARTNER LU The local LU sent out a random number to another LU as part of the handshaking process of establishing a session. That same number then came in from a third LU for the local LU to encrypt. It is a coincidence that the same number is chosen; the number is 64 bits of random data.

It may be that an unauthorized user is attempting to steal the encrypted response.

8(8)

SESSION KEY NOT DEFINED FOR PARTNER LU The local LU had VERIFY=OPTIONAL coded on its APPL statement. There was a password defined in the local LU's RACF profile for the LU-LU

pair, indicating that session level verification should be used on all sessions between the two LUs. However, the partner LU tried to start a session without using session level verification.

9(9)

SESSION KEY NOT DEFINED FOR THIS LU The local LU had VERIFY=OPTIONAL coded on its APPL statement. No password was defined in the local LU's RACF profile for the LU-LU pair, indicating that session level verification may not be used to establish sessions with this LU. However, the partner LU tried to establish a session using session level verification.

10(A)

SNA SECURITY-RELATED PROTOCOL ERROR The LU trying to establish a connection is not responding correctly according to the handshaking protocol.

11(B)

PROFILE CHANGE DURING VERIFICATION The handshaking was attempted, but it is evident that one of the LU's profiles (specifically the session key) changed in the middle of the handshaking, making its success impossible.

12(C)

EXPIRED SESSION KEY The session key in one or both of the APPCLU profiles has expired.

Event 27(1B): GENERAL AUDITING

This event is logged by RACROUTE REQUEST=AUDIT,EVENT='GENERAL'. RACF does not make any authority checks for this event.

The explanations of the event code qualifiers for Event 27 are:

0 - 99 GENERAL AUDIT RECORD WRITTEN

Qualifiers 0 to 99 can be used for Event 27. These qualifiers are installation defined.

Event 28(1C)–58(3A): z/OS UNIX EVENT TYPES

Events 28 through 58 apply to z/OS UNIX. The following qualifier codes are used for each event:

28(1C)**DIRECTORY SEARCH****0(0)**

Access allowed

1(1)

Not authorized to search directory

2(2)

Security label failure

29(1D)**CHECK ACCESS TO DIRECTORY****0(0)**

Access allowed

1(1)

Caller does not have requested access authority

2(2)

Security label failure

30(1E)**CHECK ACCESS TO FILE****0(0)**

Access allowed

1(1)

Caller does not have requested access authority

Event codes

2(2)

Security label failure

31(1F)

CHAUDIT

0(0)

File's audit options changed

1(1)

Caller does not have authority to change user audit options of specified file

2(2)

Caller does not have authority to change auditor audit options

3(3)

Security label failure

32(20)

CHDIR

0(0)

Current working directory changed

Failures logged as directory search event types

33(21)

CHMOD

0(0)

File's mode changed

1(1)

Caller does not have authority to change mode of specified file

2(2)

Security label failure

34(22)

CHOWN

0(0)

File's owner or group owner changed

1(1)

Caller does not have authority to change owner or group owner of specified file

2(2)

Security label failure

35(23)

CLEAR SETID BITS FOR FILE

0(0)

S_ISUID, S_ISGID, and S_ISVTX bits changed to zero (write)

No failure cases

36(24)

EXEC WITH SETUID/SETGID

0(0)

Successful change of UIDs and GIDs

No failure cases

37(25)

GETPSENT

- 0(0)**
Access allowed
- 1(1)**
Not authorized to access specified process
- 38(26)**
INITIALIZE z/OS UNIX PROCESS (DUB)
 - 0(0)**
z/OS UNIX process successfully initiated
 - 1(1)**
User not defined as a z/OS UNIX user (no user profile or no OMVS segment)
 - 2(2)**
User incompletely defined as a z/OS UNIX user (no UID in user profile)
 - 3(3)**
User's current group has no GID
- 39(27)**
z/OS UNIX PROCESS COMPLETION (UNDUB)
 - 0(0)**
Process completed
 - No failure cases
- 40(28)**
KILL
 - 0(0)**
Access allowed
 - 1(1)**
Not authorized to access specified process
 - 2(2)**
Security label failure
- 41(29)**
LINK
 - 0(0)**
New link created
 - ***
Failures logged as directory search or check access event types
- 42(2A)**
MKDIR
 - 0(0)**
Directory successfully created
 - ***
Failures logged as directory search or check access event types
- 43(2B)**
MKNOD
 - 0(0)**
Successful creation of a node
 - ***
Failures logged as directory search or check access event types
- 44(2C)**
MOUNT FILE SYSTEM

Event codes

0(0)

Successful mount

Failures logged as ck_priv event type

45(2D)

OPEN (NEW FILE)

0(0)

File successfully created

Failures logged as directory search or check access event types

46(2E)

PTRACE

0(0)

Access allowed

1(1)

Not authorized to access specified process

2(2)

Security label failure

47(2F)

RENAME

0(0)

Rename successful

Failures logged as directory search or check access event types

48(30)

RMDIR

0(0)

Successful rmdir

Failures logged as directory search or check access event types

49(31)

SETEGID

0(0)

Successful change of effective GID

1(1)

Not authorized to setegid

50(32)

SETEUID

0(0)

Successful change of effective UID

1(1)

Not authorized to seteuid

51(33)

SETGID

0(0)

Successful change of GIDs

1(1)

Not authorized to setgid

52(34)	SETUID
0(0)	Successful change of UIDs
1(1)	Not authorized to setuid
53(35)	SYMLINK
0(0)	Successful symlink
*	Failures logged as directory search or check access event types
54(36)	UNLINK
0(0)	Successful unlink
*	Failures logged as directory search or check access event types
55(37)	UNMOUNT FILE SYSTEM
0(0)	Successful unmount
*	Failures logged as ck_priv event type
56(38)	CHECK FILE OWNER
0(0)	User is the owner
1(1)	User is not the owner
2(2)	Security label failure
57(39)	CK_PRIV
0(0)	User is authorized
1(1)	User not authorized to use requested function
58(3A)	OPEN SUBSIDIARY TTY
0(0)	Access allowed
1(1)	Not authorized to access specified process

Event 59(3B): RACLINK EVENT TYPES

The explanations of the event code qualifiers for Event 59 are:

Event codes

0(0)

No violation detected

1(1)

Insufficient authority

2(2)

Keyword violation detected

3(3)

Association already defined

4(4)

Association already approved

5(5)

Association does not match

6(6)

Association does not exist

7(7)

Invalid password or revoked user ID

Event 60(3C)–62(3E): z/OS UNIX XPG4 EVENT TYPES

60(3C)

CHECK IPC ACCESS

0(0)

Access allowed

1(1)

Caller does not have requested access authority

2(2)

Security label failure

61(3D)

MAKE ISP

0(0)

Successful creation of ISP

1(1)

Security label failure

62(3E)

R_IPC CONTROL

0(0)

Access allowed

1(1)

Caller does not have requested access authority

2(2)

Security label failure

Event 63(3F): z/OS UNIX SETGROUPS EVENT TYPE

0(0)

Successful

1(1)

Not authorized

Event 64(40): X/OPEN SINGLE UNIX SPECIFICATION EVENT TYPES

64(40)

CHECK OWNER TWO FILES

0(0)

User is the owner

1(1)

User is not the owner

2(2)

Security label failure

Event 65(41): z/OS UNIX PASSING OF ACCESS RIGHTS EVENT TYPES

65(41)

R_AUDIT

0(0)

Successful r_audit

No failure cases

Event 66(42)–67(43): CERTIFICATE EVENT TYPES

66(42)

RACDCERT

0(0)

No violation detected

1(1)

Insufficient authority

67(43)

initACEE

0(0)

Successful certificate registration

1(1)

Successful certificate deregistration

2(2)

Insufficient authority to register a certificate

3(3)

Insufficient authority to deregister a certificate

4(4)

No user ID found for certificate

5(5)

Certificate is not trusted

6(6)

Successful CERTAUTH certificate registration

7(7)

Insufficient authority to register the CERTAUTH certificate

8(8)

Client security label not equivalent to server's

9(9)

Invalid use of reserved user ID

Event codes

10(A)

No RACF userID found for distributed identity

11(B)

Successful IDT generated from ACEE

12(C)

Failed attempting to generate IDT from ACEE

Event 68(44): GRANT OF INITIAL KERBEROS TICKET

68(44)

Kerberos

0(0)

Success

1(1)

Failure

Event 69(45): R_PKIServ GENCERT

69(45)

RPKIGENC

0(0)

Successful GENCERT request

1(1)

Insufficient authority for GENCERT

2(2)

Successful REQCERT request

3(3)

Insufficient authority for REQCERT

4(4)

Successful GENRENEW request

5(5)

Insufficient authority for GENRENEW

6(6)

Successful REQRENEW request

7(7)

Insufficient authority for REQRENEW

8(8)

Successful PREREGISTER request

9(9)

Insufficient authority for PREREGISTER

Event 70(46): R_PKIServ EXPORT

70(46)

RPKIEXPT

0(0)

Successful certificate EXPORT request

1(1)

Unsuccessful certificate EXPORT request due to insufficient authority

2(2)

Incorrect pass phrase specified for EXPORT

Event 71(47): POLICY DIRECTOR ACCESS CONTROL DECISION

71(47)

PDACCESS

This event is reserved for use by Policy Director Authorization Services.

0(0)

Authorized

1(1)

Not authorized but permitted because of warning mode

2(2)

Not authorized due to insufficient traverse authority but permitted because of warning mode

3(3)

Not authorized due to time-of-day check but permitted because of warning mode

4(4)

Not authorized

5(5)

Not authorized due to insufficient traverse authority

6(6)

Not authorized due to time-of-day check

Event 72(48): R_PKIServ QUERY

72(48)

RPKIREAD

0(0)

Successful admin QUERY or DETAILS request

1(1)

Insufficient authority for admin QUERY or DETAILS

2(2)

Successful VERIFY request

3(3)

Insufficient authority for VERIFY

4(4)

Incorrect VERIFY certificate, no record found for this certificate

Event 73(49): R_PKIServ UPDATEREQ

73(49)

RPKIUPDR

0(0)

Successful admin UPDATEREQ request

1(1)

Insufficient authority for admin UPDATEREQ

Event 74(4A): R_PKIServ UPDATECERT

74(4A)

RPKIUPDC

0(0)

Successful admin UPDATECERT request

1(1)

Insufficient authority for admin UPDATECERT

Event codes

2(2)

Successful REVOKE request

3(3)

Insufficient authority for REVOKE

Event 75(4B): CHANGE FILE ACL

75(4B)

SETFACL

0(0)

ACL entry added, changed, or deleted

1(1)

Caller does not have authority to change ACL of specified file

2(2)

Security label failure

Event 76(4C): REMOVE FILE ACL

76(4C)

DELFACL

0(0)

Entire ACL deleted

1(1)

Caller does not have authority to remove ACL of specified file

2(2)

Security label failure

Event 77(4D): SET FILE SECURITY LABEL

77(4D)

SETFSECL

0(0)

Security label change

1(1)

Not authorized to change security label

Event 78(4E): SET WRITE-DOWN PRIVILEGE

78(4E)

WRITEDWN

0(0)

Requested function successful

1(1)

Not authorized to IRR.WRITEDOWN.BYUSER

Event 79(4F): CRL PUBLICATION

79(4F)

PKIDPUBR

0(0)

Successful publication of revocation information

Event 80(50): R_PKIServ RESPOND**80(50)****RPKIRESP****0**

Successful RESPOND request

1

Insufficient authority for RESPOND

Event 81(51): PassTicket Evaluation**81****0**

Successful request

1

Request failed

Event 82(52): PassTicket Generation**82****0**

Successful generation

1

Generation request failed

Event 83(53): R_PKIServ SCEPREQ**83(53)****RPKISCEP****0**

Successful AutoApprove PKCSReq request

1

Successful AdminApprove PKCSReq request

2

Successful GetCertInitial request

3

Rejected PKCSReq or GetCertInitial request

4

Incorrect SCEP transaction ID specified for GetCertInitial

5

Insufficient authority for SCEPREQ

Event 84(54): R_Datalib RDATAUPD**84(54)****RDATAUPD****0**

Successful NewRing

1

Not authorized to call NewRing

Event codes

- 2** Successful DataPut
- 3** Not authorized to call DataPut
- 4** Successful DataRemove
- 5** Not authorized to call DataRemove
- 6** Successful DelRing
- 7** Not authorized to call DelRing

Event 85(55): PKIAURNW

85(55)

PKIAURNW

0

Successful renewal of a PKI Services issued certificate

Event 86(56): R_PgmSignVer

86(56)

R_PgmSignVer

0

Successful signature verification

1

Signature appears valid but root CA certificate not trusted

2

Module signature failed verification

3

Module certificate chain incorrect

4

Signature required but module not signed

5

Signature required but signature has been removed

6

Program verification module not loaded. Program verification was not available when attempt was made to load this program.

7

The Algorithmic self test failed while verifying the program verification module.

Event 87(57): RACMAP

87(57)

RACMAP

0(0)

No violation detected

1(1)

Insufficient authority (no update to RACF database)

Event 88(58): AUTOPROF**88(58)****AUTOPROF****0**

Successful profile modification

Event 89(59): RPKIQREC**89(59)****RPKIQREC****0**

Successful user QRECOVER request

1

Insufficient authority for user QRECOVER

Event 90(5A): PKIGENC**90(5A)****PKIGENC****0**

Successful GENCERT request

Appendix F. Accessibility

Accessible publications for this product are offered through [IBM Documentation for z/OS \(www.ibm.com/docs/en/zos\)](http://www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS documentation see [How to Send Feedback to IBM](#) to leave documentation feedback.

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

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

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

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

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

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Corporation
Site Counsel
2455 South Road*

Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: [IBM Lifecycle Support for z/OS \(www.ibm.com/software/support/systemsz/lifecycle\)](http://www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Programming interface information

This book is intended to help you with the following tasks:

- Install, maintain, or modify RACF using the macros provided by RACF.
- Code the interfaces used to invoke RACF using the RACF ISPF panels.

This publication primarily documents intended programming interfaces that allow you to write programs to obtain the services of RACF.

This publication also documents information that is NOT intended to be used as programming interfaces of RACF. This information is identified where it occurs, either by an introductory statement to a topic or by the following marking:

NOT Programming Interface Information

End NOT Programming Interface Information

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at [Copyright and Trademark information \(www.ibm.com/legal/copytrade.shtml\)](http://www.ibm.com/legal/copytrade.shtml).

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux[®] is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Index

A

ACCESS
 event qualifiers [182](#)
 record extension [182](#)
access control list
 event qualifiers [318](#), [321](#)
 record extension format [318](#), [321](#)
access rights
 event qualifiers [303](#)
 record extension [303](#)
accessibility
 contact IBM [681](#)
ACEE keyword
 on ICHEINTY macro [496](#)
ACTION keyword
 on ICHNCONV macro [17](#)
 on ICHRFRTB macro [20](#)
ACTIONS keyword
 on ICHEINTY macro [498](#)
ADDGROUP
 event qualifiers [195](#)
 record extension [195](#)
ADDSD
 event qualifiers [194](#)
 record extension [194](#)
ADDUSER
 event qualifiers [197](#)
 record extension [197](#)
ADDVOL
 event qualifiers [185](#)
 record extension [185](#)
algorithm
 input data [425](#)
 PassTicket generator [421](#)
 secured signon session key generator [435](#)
ALTDSD
 event qualifiers [199](#)
 record extension [199](#)
ALTGROUP
 event qualifiers [200](#)
 record extension [200](#)
ALTUSER
 event qualifiers [202](#)
 record extension [202](#)
APPC session establishment records [221](#)
APPCLU
 event qualifiers [221](#)
 record extension [221](#)
application key
 description [426](#)
assistive technologies [681](#)
AUTOPROF
 event qualifiers [339](#)
 record extension format [339](#)

C

CASE keyword
 on ICHERCDE macro [2](#)
CDT (class descriptor table)
 classes supplied by IBM [523](#)
 syntax of the ICHERCDE macro [1](#)
 sysplex communication [1](#)
 when the RACF database is shared [1](#)
CHAIN keyword
 on ICHEINTY macro [497](#)
change audit
 event qualifiers [231](#)
 record extension [231](#)
change directory
 event qualifiers [234](#)
 record extension [234](#)
change file mode
 event qualifiers [236](#)
 record extension [236](#)
change file ownership
 event qualifiers [239](#)
 record extension [239](#)
check directory access
 event qualifiers [227](#)
 record extension [227](#)
check file access
 event qualifiers [229](#)
 record extension format [229](#)
check file owner
 event qualifiers [284](#)
 record extension [284](#)
check privilege
 event qualifiers [286](#)
 record extension [286](#)
CKOWN2
 event qualifiers [301](#)
 record extension [301](#)
class descriptor table (CDT)
 dynamic [1](#)
 static [1](#)
 sysplex communication [1](#)
 when the RACF database is shared [1](#)
CLASS keyword
 on ICHEINTY macro [494](#)
 on ICHERCDE macro [2](#)
 on ICHRFRTB macro [20](#)
clear SETID bits
 event qualifiers [241](#)
 record extension [241](#)
combination field definitions
 definition [609](#)
 format [609](#)
COND keyword
 on ICHETEST macro [506](#)
 on ICHNCONV SELECT macro [11](#)
CONNECT

- CONNECT (*continued*)
 - event qualifiers [203](#)
 - record extension [203](#)
- connect template
 - contents [629](#)
- contact
 - z/OS [681](#)
- conversion rules for database unload [366](#)
- converting R_admin extract requests, IRRXUTIL [465](#)
- CRL publication
 - event qualifiers [326](#)
 - record extension format [326](#)

D

- data set
 - record formats [392](#)
- data set template
 - contents [631](#)
- data unload utility [169](#)
- database
 - locating or updating a profile with ICHEINTY [490](#)
 - using macros to update the profiles [489](#)
- database profile
 - storage requirements [609](#), [611](#)
- database unload utility (IRRDDBU00)
 - conversion rules [366](#)
 - record formats produced [367](#)
- DATAMAP keyword
 - on ICHEINTY macro [497](#)
- date conversion routine [29](#)
- date fields [608](#)
- DATEFMT keyword
 - on ICHEINTY macro [501](#)
- DEFINE
 - event qualifiers [192](#)
 - record extension [192](#)
- DEFINE keyword
 - on ICHNCONV macro [11](#)
- DELDSD
 - event qualifiers [205](#)
 - record extension [205](#)
- DELFACL
 - event qualifiers [321](#)
 - record extension format [321](#)
- DELGROUP
 - event qualifiers [206](#)
 - record extension [206](#)
- DELRES
 - event qualifiers [189](#)
 - record extension [189](#)
- DELUSER
 - event qualifiers [208](#)
 - record extension [208](#)
- DELVOL
 - event qualifiers [191](#)
 - record extension [191](#)
- DFTRETC keyword
 - on ICHERCDE macro [2](#)
- DFTUACC keyword
 - on ICHERCDE macro [3](#)
- directory search
 - event qualifiers [224](#)
 - record extension [224](#)

dynamic class descriptor table [1](#)

E

- ENCRYPT keyword
 - on ICHEACTN macro [510](#)
 - on ICHECTEST macro [507](#)
- END keyword
 - on ICHNCONV macro [18](#)
- ENTRY keyword
 - on ICHEINTY macro [494](#)
- environment service, IRRENS00 [437](#)
- EQUALMAC keyword
 - on ICHERCDE macro [3](#)
- event codes
 - for SMF data unload utility [173](#)
 - for type 80 SMF records [41](#)
 - qualifier descriptions [655](#)
 - qualifiers
 - for type 80 SMF records [41](#)
- event qualifiers
 - ACCESS [182](#)
 - access control list [318](#), [321](#)
 - access rights [303](#)
 - ADDGROUP [195](#)
 - ADDSD [194](#)
 - ADDUSER [197](#)
 - ADDVOL [185](#)
 - ALTDSD [199](#)
 - ALTGROUP [200](#)
 - ALTUSER [202](#)
 - APPCLU [221](#)
 - AUTOPROF [339](#)
 - change audit [231](#)
 - change directory [234](#)
 - change file mode [236](#)
 - change file ownership [239](#)
 - check directory access [227](#)
 - check file access [229](#)
 - check file owner [284](#)
 - check privilege [286](#)
 - CKOWN2 [301](#)
 - clear SETID bits [241](#)
 - CONNECT [203](#)
 - CRL publication [326](#)
 - DEFINE [192](#)
 - DELDSD [205](#)
 - DELFACL [321](#)
 - DELGROUP [206](#)
 - DELRES [189](#)
 - DELUSER [208](#)
 - DELVOL [191](#)
 - directory search [224](#)
 - EXEC SETGID [243](#)
 - EXEC SETUID [243](#)
 - GETPSENT [245](#)
 - initACEE [306](#)
 - IPCCHK [292](#)
 - IPCCTL [296](#)
 - IPCGET [294](#)
 - JOBINIT [177](#)
 - KILL [250](#)
 - LINK [252](#)
 - MKDIR [254](#)

event qualifiers (*continued*)

- MKNOD [257](#)
- mount file system [260](#)
- Network Authentication Service [308](#)
- open subsidiary TTY [287](#)
- OPENFILE [262](#)
- PassTicket evaluation [327](#)
- PassTicket generation [329](#)
- PASSWORD [209](#)
- PERMIT [211](#)
- PGMVERIFY [336](#)
- PKIAURNW [335](#)
- PKIDPUBR [326](#)
- PKIGENC [342](#)
- Policy Director [312](#)
- PRLIMIT [343](#)
- PTCREATE [329](#)
- PTEVAL [327](#)
- PTRACE [265](#)
- RACDCERT [304](#)
- RACLINK command [289](#)
- RACMAP [337](#)
- RALTER [212](#)
- RDATAUPD [333](#)
- RDEFINE [214](#)
- RDELETE [215](#)
- REMOVE [217](#)
- rename file [267](#)
- RENAMEDS [187](#)
- RMDIR [269](#)
- RPKIEPT [310](#)
- RPKIGENC [308](#)
- RPKIREAD [313](#)
- RPKIRESP [326](#)
- RPKISCEP [331](#)
- RPKIUPDC [317](#)
- RPKIUPDR [315](#)
- RVARY [220](#)
- security event [345](#)
- set file security label [323](#)
- set write-down privilege [324](#)
- SETEGID [271](#)
- SETEUID [273](#)
- SETFACL [318](#)
- SETFSECL [323](#)
- SETGID [275](#)
- SETGROUP [299](#)
- SETROPTS [218](#)
- SETUID [276](#)
- SYMLINK [278](#)
- UNLINK [280](#)
- unmount file system [282](#)
- WRITEDWN [324](#)
- z/OS UNIX process completion [249](#)
- z/OS UNIX process initialization [247](#)
- event qualifiers for general events [223](#)
- EVENT variable
 - on ICHNCONV SELECT macro [14](#)
- EXEC SETGID
 - event qualifiers [243](#)
 - record extension [243](#)
- EXEC SETUID
 - event qualifiers [243](#)

EXEC SETUID (*continued*)

- record extension [243](#)
- extended-length relocate section
 - variable data for type 80 SMF records [65](#)

F

- FIELD keyword
 - on ICHEACTN macro [508](#)
 - on ICHETEST macro [506](#)
- fields
 - character [609](#)
 - date [608](#)
 - integer [609](#)
 - time [609](#)
- FINAL keyword
 - on ICHNCONV macro [18](#)
- FIRST keyword
 - on ICHERCDE macro [3](#)
- FLDATA keyword
 - on ICHEACTN macro [508](#)
 - on ICHETEST macro [506](#)
- FLDEF keyword
 - on ICHEINTY macro [500](#)

G

- G variable
 - on ICHNCONV SELECT macro [13](#)
- general event
 - record extension [223](#)
- general resource
 - fields in the profile [636](#)
 - record formats [397](#)
- general template
 - contents [636](#)
- generator algorithm for PassTicket [421](#)
- generator algorithm for secured signon session key [435](#)
- GENERIC keyword
 - on ICHEINTY macro [500](#)
 - on ICHERCDE macro [4](#)
- GENLIST keyword
 - on ICHERCDE macro [4](#)
- GETPSENT
 - event qualifiers [245](#)
 - record extension [245](#)
- GQ variable
 - on ICHNCONV SELECT macro [13](#)
- group
 - maximum number of users in [610](#)
 - record formats [367](#)
 - template for database [611](#)
- GROUP keyword
 - on ICHERCDE macro [4](#)
- group profile
 - description of the fields [611](#)
- group template
 - contents [611](#)

H

- header
 - for SMF records [31](#)

I

ICH408I message [31](#)
ICHEACTN macro
 description [508](#)
 examples of [516](#)
 format of the user work area DATAMAP=NEW [511](#)
 format of the user work area DATAMAP=OLD [514](#)
 syntax [508](#)
 using it to retrieve data with DATAMAP=NEW [511](#)
 using to alter data [513](#), [516](#)
 using to retrieve data when DATAMAP=OLD [514](#)
ICHEINTY macro
 examples of [516](#)
 format of the user work area when INDEX=MULTIPLE is not specified [498](#)
 format of the user work area when INDEX=MULTIPLE is specified [499](#)
 return codes [502](#)
 syntax [490](#)
ICHERCDE macro
 class descriptor table supplied by IBM [523](#)
 description [1](#)
 syntax [1](#)
ICHETEST macro
 considerations when using [507](#)
 description [505](#)
 examples of [516](#)
 syntax [505](#)
ICHNCONV ACTION macro
 description [17](#)
 syntax [17](#)
ICHNCONV DEFINE macro
 description [10](#)
 syntax [10](#)
ICHNCONV END macro
 description [18](#)
 syntax [18](#)
ICHNCONV FINAL macro
 description [18](#)
 syntax [18](#)
ICHNCONV macro
 description [10](#)
 example of coding [18](#)
ICHNCONV SELECT macro
 description [11](#)
 syntax [11](#)
ICHRFR01 module
 syntax of the ICHRFRTB macro [19](#)
ICHRFRTB macro
 description [19](#)
 syntax [20](#)
ICHRRCDE module
 generating entries [1](#)
ICHRRCDX table [523](#)
ID keyword
 on ICHERCDE macro [5](#)
INDEX keyword
 on ICHEINTY macro [501](#)
initACEE
 event qualifiers [306](#)
 record extension [306](#)
initialization record (type 81) [136](#)

initialize z/OS UNIX
 process
 event qualifiers [247](#)
 record extension [247](#)
IPCCHK
 event qualifiers [292](#)
 record extension [292](#)
IPCCTL
 event qualifiers [296](#)
 record extension [296](#)
IPCGET
 event qualifiers [294](#)
 record extension [294](#)
IRRADU00
 record format [169](#)
IRRDBU00 utility
 record types [355](#)
IRRDCR00 module [29](#)
IRRENS00 environment service [437](#)
IRRGNT00 translate service [461](#)
IRRPNL00 module [25](#)
IRRUTIL
 class descriptor entry data [485](#)

J

JOBINIT
 event qualifiers [177](#)
 record extension [177](#)

K

keyboard
 navigation [681](#)
 PF keys [681](#)
 shortcut keys [681](#)
KEYQUAL keyword
 on ICHERCDE macro [5](#)
KILL
 event qualifiers [250](#)
 record extension [250](#)

L

LINK
 event qualifiers [252](#)
 record extension [252](#)

M

macros
 customization [1](#)
 ICHEACTN [508](#)
 ICHEINTY [490](#)
 ICHERCDE macro [1](#)
 ICHETEST [505](#)
 ICHNCONV [10](#)
 ICHRFRTB macro [19](#)
 that are part of RACF [1](#)
 maximum number of users in group [610](#)
MAXLENX keyword
 on ICHERCDE macro [6](#)
MAXLNTH keyword

MAXLNTH keyword (*continued*)

on ICHERCDE macro [6](#)

MEMBER keyword

on ICHERCDE macro [6](#)

messages

ICH408I [31](#)

MF keyword

on ICHEACTN macro [510](#)

on ICHEINTY macro [501](#)

on ICHETEST macro [507](#)

MKDIR

event qualifiers [254](#)

record extension [254](#)

MKNOD

event qualifiers [257](#)

record extension [257](#)

mount file system

event qualifiers [260](#)

record extension [260](#)

N

NAME keyword

on ICHNCONV DEFINE macro [11](#)

NAMETYPE variable

on ICHNCONV SELECT macro [14](#)

naming convention [10](#)

naming convention table

example of coding [18](#)

syntax of the ICHNCONV macro [10](#)

navigation

keyboard [681](#)

Network Authentication Service

event qualifiers [308](#)

record extension [308](#)

NEWNAME keyword

on ICHEINTY macro [499](#)

NEWNAMX keyword

on ICHEINTY macro [499](#)

NEXT keyword

on ICHNCONV END macro [18](#)

O

OLDVOL variable

on ICHNCONV SELECT macro [16](#)

open subsidiary TTY

event qualifiers [287](#)

record extension [287](#)

OPENFILE

event qualifiers [262](#)

record extension [262](#)

OPER keyword

on ICHERCDE macro [6](#)

operand on COND keyword

on ICHNCONV SELECT macro [17](#)

operation value

on ICHEINTY macro [490](#)

operator on COND keyword

on ICHNCONV SELECT macro [16](#)

OPTIONS keyword

on ICHEINTY macro [500](#)

OTHER keyword

OTHER keyword (*continued*)

on ICHERCDE macro [6](#)

P

panel driver interface [21](#)

PassTicket

definition [419](#)

generating a session key with [433](#)

generating and evaluating

callable service interface [419](#)

incorporating PassTicket generator algorithm in a program [421](#)

Java interface [419](#)

secured signon interface [420](#)

generator algorithm

description of [421](#)

secured signon session key generator algorithm

description of [435](#)

PassTicket evaluation

event qualifiers [327](#)

record extension format [327](#)

PassTicket generation

event qualifiers [329](#)

record extension format [329](#)

password

PassTicket as an alternative for [419](#)

PASSWORD

event qualifiers [209](#)

record extension [209](#)

PERMIT

event qualifiers [211](#)

record extension [211](#)

PGMVERIFY

event qualifiers [336](#)

record extension format [336](#)

PKIAURNW

event qualifiers [335](#)

record extension format [335](#)

PKIDPUBR

event qualifiers [326](#)

record extension format [326](#)

PKIGENC

event qualifiers [342](#)

record extension [342](#)

Policy Director

event qualifiers [312](#)

record extension [312](#)

POSIT keyword

on ICHERCDE macro [7](#)

PRLIMIT

event qualifiers [343](#)

record extension [343](#)

processing record [31](#)

processing record for auditing data sets [145](#)

PROFDEF keyword

on ICHERCDE macro [9](#)

profile

contents of a data set profile [631](#)

contents of a general resource profile [636](#)

contents of a group profile [611](#)

contents of a user profile [614](#)

in database [608](#)

locating or updating with ICHEINTY [490](#)

- profile (*continued*)
 - repeat groups [608](#)
 - retrieving and altering data with ICHEACTN [508](#)
 - testing for conditions with ICHETEST [505](#)
 - updating on the RACF database with macros [489](#)
- profile name
 - PTKTDATA class [426](#)
- profile name list service routine [25](#)
- PTCREATE
 - event qualifiers [329](#)
 - record extension format [329](#)
- PTEVAL
 - event qualifiers [327](#)
 - record extension format [327](#)
- PTKTDATA
 - class profile name [426](#)
- PTRACE
 - event qualifiers [265](#)
 - record extension [265](#)

Q

- QCT variable
 - on ICHNCONV SELECT macro [14](#)
- QUAL variable
 - on ICHNCONV SELECT macro [14](#)

R

- RACDCERT
 - event qualifiers [304](#)
 - record extension [304](#)
- RACF
 - panel driver interface [21](#)
 - SMF records [31](#)
- RACF commands
 - SMF command-related data [81](#)
- RACF database
 - connect template [629](#)
 - general template [636](#)
 - group template [611](#)
 - reserved template [652](#)
 - user template [614](#)
 - using macros to update the profiles [489](#)
- RACF macros
 - ICHNCONV
 - FINAL [18](#)
 - ICHNCONV macro
 - ACTION [17](#)
 - DEFINE [10](#)
 - END [18](#)
 - SELECT [11](#)
- RACF PassTicket [419](#)
- RACF profile name list service routine [25](#)
- RACF report writer
 - types of records it reformats [31](#)
- RACF router table
 - generating entries for [19](#)
- RACGPID on COND keyword
 - on ICHNCONV SELECT macro [16](#)
- RACGPID3 on COND keyword
 - on ICHNCONV SELECT macro [16](#)
- RACLINK command

- RACLINK command (*continued*)
 - event qualifiers [289](#)
 - record extension [289](#)
- RACLIST keyword
 - on ICHERCDE macro [9](#)
- RACLREQ keyword
 - on ICHERCDE macro [9](#)
- RACMAP
 - event qualifiers [337](#)
 - record extension format [337](#)
- RACROUTE macro
 - relationship to ICHRFRTB macro [19](#)
- RACUID on COND keyword
 - on ICHNCONV SELECT macro [16](#)
- RACUID3 on COND keyword
 - on ICHNCONV SELECT macro [16](#)
- RACVAR function for REXX execs
 - using [521](#)
- RALTER
 - event qualifiers [212](#)
 - record extension [212](#)
- RBA keyword
 - on ICHEINTY macro [500](#)
- RDATAUPD
 - event qualifiers [333](#)
 - record extension format [333](#)
- RDEFINE
 - event qualifiers [214](#)
 - record extension [214](#)
- RDELETE
 - event qualifiers [215](#)
 - record extension [215](#)
- record dependent section
 - of SMF process records [160](#)
 - of SMF status records [161](#)
- record extension
 - ACCESS [182](#)
 - access control list [318](#), [321](#)
 - access rights [303](#)
 - ADDGROUP [195](#)
 - ADDSD [194](#)
 - ADDUSER [197](#)
 - ADDVOL [185](#)
 - ALTDSD [199](#)
 - ALTGROUP [200](#)
 - ALTUSER [202](#)
 - APPCLU [221](#)
 - AUTOPROF [339](#)
 - change audit [231](#)
 - change directory [234](#)
 - change file mode [236](#)
 - change file ownership [239](#)
 - check file access [229](#)
 - check file owner [284](#)
 - check privilege [286](#)
 - CKOWN2 [301](#)
 - clear SETID bits [241](#)
 - CONNECT [203](#)
 - CRL publication [326](#)
 - DEFINE [192](#)
 - DELDSD [205](#)
 - DELFACL [321](#)
 - DELGROUP [206](#)
 - DELRES [189](#)

record extension (*continued*)

- [DELUSER 208](#)
- [DELVOL 191](#)
- [directory search 224](#)
- [EXEC SETGID 243](#)
- [EXEC SETUID 243](#)
- [general event 223](#)
- [GETPSENT 245](#)
- [initACEE 306](#)
- [initialize z/OS UNIX process 247](#)
- [IPCCHK 292](#)
- [IPCCTL 296](#)
- [IPCGET 294](#)
- [JOBINIT 177](#)
- [KILL 250](#)
- [LINK 252](#)
- [MKDIR 254](#)
- [MKNOD 257](#)
- [mount file system 260](#)
- [Network Authentication Service 308](#)
- [open subsidiary TTY 287](#)
- [OPENFILE 262](#)
- [PassTicket evaluation 327](#)
- [PassTicket generation 329](#)
- [PASSWORD 209](#)
- [PERMIT 211](#)
- [PGMVERIFY 336](#)
- [PKIAURNW 335](#)
- [PKIDPUBR 326](#)
- [PKIGENC 342](#)
- [Policy Director 312](#)
- [PRLIMIT 343](#)
- [PTCREATE 329](#)
- [PTEVAL 327](#)
- [PTRACE 265](#)
- [RACDCERT 304](#)
- [RACLINK command 289](#)
- [RACMAP 337](#)
- [RALTER 212](#)
- [RDATAUPD 333](#)
- [RDEFINE 214](#)
- [RDELETE 215](#)
- [REMOVE 217](#)
- [rename file 267](#)
- [RENAMEDS 187](#)
- [RMDIR 269](#)
- [RPKIEXPT 310](#)
- [RPKIGENC 308](#)
- [RPKIQREC 341](#)
- [RPKIREAD 313](#)
- [RPKIRESP 326](#)
- [RPKISCEP 331](#)
- [RPKIUPDC 317](#)
- [RPKIUPDR 315](#)
- [RVARY 220](#)
- [security event 345](#)
- [set file security label 323](#)
- [set write-down privilege 324](#)
- [SETEGID 271](#)
- [SETFSECL 323](#)
- [SETGID 275](#)
- [SETGROUP 299](#)
- [SETROPTS 218](#)
- [SETUID 273, 276](#)

record extension (*continued*)

- [SYMLINK 278](#)
- [UNLINK 280](#)
- [unmount file system 282](#)
- [WRITEDWN 324](#)
- [z/OS UNIX process completion 249](#)
- record formats
 - [data set 392, 397](#)
 - [for database unload utility 367](#)
 - [group 367](#)
 - [user 370](#)
- records
 - SMF
 - [reformatted process 156](#)
 - [reformatted status 161](#)
 - [type 80 31](#)
 - [type 81 136](#)
 - [type 83 145](#)
 - [reformatted process records](#)
 - [format of 156](#)
 - [record dependent section 160](#)
 - [reformatted SMF records](#)
 - [description 156](#)
 - [types of 31](#)
 - [reformatted status records](#)
 - [format of 161](#)
 - [record dependent section 161](#)
- RELEASE keyword
 - [on ICHEACTN macro 510](#)
 - [on ICHEINTY macro 495](#)
 - [on ICHETEST macro 507](#)
- relocate section
 - [for reformatted process records 160](#)
 - [for reformatted status records 167](#)
 - [variable data for type 80 SMF records 56](#)
- REMOVE
 - [event qualifiers 217](#)
 - [record extension 217](#)
- rename file
 - [event qualifiers 267](#)
 - [record extension 267](#)
- RENAMEDS
 - [event qualifiers 187](#)
 - [record extension 187](#)
- repeat groups
 - [in database 608](#)
 - [when using ICHETEST macro 507](#)
- report writer
 - [reformatted SMF records 156](#)
 - [types of records it reformats 31](#)
- REQSTOR keyword
 - [on ICHRFRTB macro 20](#)
- return codes
 - [from ICHEINTY macro 502](#)
- REXX RACVAR function
 - [using 521](#)
- REXX stem variables [465](#)
- REXX stem variables created by IRRXUTIL
 - [profile and SETROPTS information 473](#)
 - [RACF subsystem and remote sharing information 476](#)
- RMDIR
 - [event qualifiers 269](#)
 - [record extension 269](#)
- router table

- router table (*continued*)
 - generating entries for [19](#)
- RPKIEXPT
 - event qualifiers [310](#)
 - record extension [310](#)
- RPKIGENC
 - event qualifiers [308](#)
 - record extension [308](#)
- RPKIQREC
 - record extension format [341](#)
- RPKIREAD
 - event qualifiers [313](#)
 - record extension [313](#)
- RPKIRESP
 - event qualifiers [326](#)
 - record extension format [326](#)
- RPKISCEP
 - event qualifiers [331](#)
 - record extension format [331](#)
- RPKIUPDC
 - event qualifiers [317](#)
 - record extension [317](#)
- RPKIUPDR
 - event qualifiers [315](#)
 - record extension [315](#)
- RUN keyword
 - on ICHEACTN macro [510](#)
 - on ICHEINTY macro [496](#)
- RVARY
 - event qualifiers [220](#)
 - record extension [220](#)
- RVRSMAC keyword
 - on ICHERCDE macro [9](#)

S

- SAF user mapping plug-in [448](#), [450](#), [454](#)
- safMappingInit() [448](#)
- safMappingLookup() [450](#)
- safMappingTerm() [454](#)
- security event
 - event qualifiers [345](#)
 - record extension [345](#)
- SEGMENT keyword
 - on ICHEINTY macro [498](#)
- SELECT keyword
 - on ICHNCONV macro [11](#)
- session key
 - creating [433](#)
- set file security label
 - event qualifiers [323](#)
 - record extension format [323](#)
- SET keyword
 - on ICHNCONV ACTION macro [17](#)
- set write-down privilege
 - event qualifiers [324](#)
 - record extension format [324](#)
- SETEGID
 - event qualifiers [271](#)
 - record extension [271](#)
- SETEUID
 - event qualifiers [273](#)
- SETFACL
 - event qualifiers [318](#)

- SETFSECL
 - event qualifiers [323](#)
 - record extension format [323](#)
- SETGID
 - event qualifiers [275](#)
 - record extension [275](#)
- SETGROUP
 - event qualifiers [299](#)
 - record extension [299](#)
- SETID bits, clear
 - event qualifiers [241](#)
 - record extension [241](#)
- SETROPTS
 - event qualifiers [218](#)
 - record extension [218](#)
- SETUID
 - event qualifiers [276](#)
 - record extension [273](#), [276](#)
- shared RACF database
 - caution for class descriptor tables [1](#)
- shortcut keys [681](#)
- SIGNAL keyword
 - on ICHERCDE macro [9](#)
- SLBLREQ keyword
 - on ICHERCDE macro [9](#)
- SMC keyword
 - on ICHEINTY macro [500](#)
- SMF record, type 80
 - table of extended-length relocate section variable data [65](#)
- SMF records
 - description of the types RACF produces [31](#)
 - header portion format [171](#)
 - reformatted
 - process records [156](#)
 - status records [161](#)
 - type 80 [31](#)
 - type 81 [31](#), [136](#)
 - type 83 [31](#), [145](#)
- SMF80DTP field
 - table of data types [56](#)
- SMF80EVQ field
 - event code qualifier descriptions [655](#)
 - table of event code qualifiers [41](#)
- SMF80EVT field
 - event code qualifier descriptions [655](#)
 - table of event codes [41](#)
- SMF80TP2 field
 - table of data types [65](#)
- static class descriptor table [1](#)
- storage requirement
 - database profiles [609](#), [611](#)
- SUBSYS keyword
 - on ICHRFRTB macro [20](#)
- summary of changes xxvii, xxviii
- supplied class descriptor table [523](#)
- SYMLINK
 - event qualifiers [278](#)
 - record extension [278](#)

T

- Table 1 (event codes and qualifiers)
 - for type 80 SMF records [41](#)

- Table 4 (command-related data)
 - for type 80 SMF records [81](#)
- tasks
 - converting RACF field names to XML tag names
 - steps for [170](#)
- templates
 - connect [629](#)
 - data set [631](#)
 - general [636](#)
 - group [611](#)
 - reserved [652](#)
 - user [614](#)
- templates for database
 - combination field definitions [609](#)
 - format of field definitions [607](#)
 - repeat groups [608](#)
- TESTS keyword
 - on ICHEACTN macro [509](#)
 - on ICHEINTY macro [498](#)
- trademarks [686](#)
- translate service, IRRGNT00 [461](#)
- translation table for RACF secured signon [429](#), [432](#)
- type 20 SMF record
 - reformatted process records [156](#)
 - when written [156](#)
- type 30 SMF record
 - reformatted process records [156](#)
 - when written [156](#)
- type 80 SMF record
 - description [31](#)
 - events written for [31](#)
 - format of [33](#)
 - list of information contained in [33](#)
 - reformatted process records [156](#)
 - reformatted status records [161](#)
 - table of command- related data [81](#)
 - table of event codes and event code qualifiers [41](#)
 - table of extended-length relocate section variable data [65](#)
 - table of relocate section variable data [56](#)
 - uses for [32](#)
- type 81 SMF record
 - class data format [351](#)
 - description [136](#)
 - events written for [136](#)
 - format of [136](#)
 - reformatted status records [161](#)
 - unloaded data format [347](#)
- type 83 SMF record
 - description [145](#)
 - events written for [145](#)
 - unloaded data format [353](#)
- TYPE keyword
 - on ICHEINTY macro [494](#)
- TYPE=END
 - on ICHRFRTB macro [20](#)

U

- U variable
 - on ICHNCONV SELECT macro [13](#)
- UNLINK
 - event qualifiers [280](#)
 - record extension [280](#)

- unloaded database records [360](#)
- unmount file system
 - event qualifiers [282](#)
 - record extension [282](#)
- UQ variable
 - on ICHNCONV SELECT macro [13](#)
- user
 - record formats [370](#)
- user interface
 - ISPF [681](#)
 - TSO/E [681](#)
- user profile
 - description of the fields [614](#)
- user template
 - contents [614](#)
- users
 - maximum number in group [610](#)

V

- V variable
 - on ICHNCONV SELECT macro [15](#)
- variable on COND keyword
 - on ICHNCONV SELECT macro
 - example of initial variable settings [13](#)
- variable on SET keyword
 - on ICHNCONV ACTION macro [17](#)
- VCT variable
 - on ICHNCONV SELECT macro [16](#)
- VOLUME keyword
 - on ICHEINTY macro [498](#)
- VOLUME variable
 - on ICHNCONV SELECT macro [15](#)

W

- WKA, WKB, and WKC variables
 - on ICHNCONV SELECT macro [16](#)
- WKAREA keyword
 - on ICHEINTY macro [498](#)
- WKSP keyword
 - on ICHEINTY macro [496](#)
- WKX, WKY, and WKZ variables
 - on ICHNCONV SELECT macro [16](#)
- WRITEDWN
 - event qualifiers [324](#)
 - record extension format [324](#)

X

- XML grammar for SMF data unload utility [170](#)

Y

- YES keyword
 - on ICHEACTN macro [510](#)

Z

- z/OS UNIX process
 - completion
 - event qualifiers [249](#)
 - record extension [249](#)

z/OS UNIX process
 initialization
 event qualifiers [247](#)
 record extension [247](#)



Product Number: 5655-ZOS

SA23-2288-70

