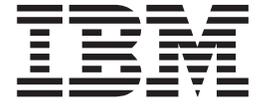
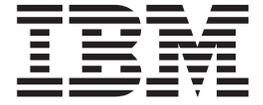


z/OS



Security Server RACF Callable Services

z/OS



Security Server RACF Callable Services

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 225.

Fourth Edition, September 2002

This is a major revision of SA22-7691-02. This edition applies to Version 1 Release 4 of z/OS (5694-A01), Version 1 Release 4 of z/OS.e (5655-G52), and to all subsequent releases and modifications until otherwise indicated in new editions.

Order documents through your IBM® representative or the IBM branch office serving your locality. Documents are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this document, or you may address your comments to the following address:

International Business Machines Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+845+432-9405

FAX (Other Countries):

Your International Access Code +1+845+432-9405

IBMLink™ (United States customers only): IBMUSM10(MHVRCFS)

Internet e-mail: mhvrcfs@us.ibm.com

World Wide Web: <http://www.ibm.com/servers/eserver/zseries/zos/webqs.html>

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this document
- Page number or topic related to your comment

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

© Copyright International Business Machines Corporation 1994, 2002. All rights reserved.

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

Contents

Tables	ix
-------------------------	-----------

About this document **xi**

Intended audience	xi
IBM systems center publications	xi
Other sources of information	xii
IBM discussion areas	xii
Internet sources	xii
To request copies of IBM publications	xiii
Where to find more information	xiv
Softcopy publications	xiv
RACF courses	xiv
Using LookAt to look up message explanations	xv
Accessing z/OS licensed documents on the Internet	xv

Summary of changes **xvii**

Chapter 1. Using the RACF Callable

Services **1**

Linkage Conventions for the Callable Services	1
Work Area (WORK)	1
File Security Packet (IFSP)	1
Security Credentials (CRED)	2
File identifiers	3
File Type and File Mode Values	4
IPC Security Packet (IISP)	5
Interprocess Communications Permission (BPXYIPCP)	6
IPC Security Credentials (CREI)	6

Chapter 2. Callable Services

Descriptions **7**

ck_access (IRRSKA00): Check Access	9
Function	9
Requirements	9
RACF authorization	9
Format	10
Parameters	10
Return and reason codes	11
Usage notes	12
Related services	12
ck_file_owner (IRRSKF00): Check File Owner	12
Function	12
Requirements	12
RACF Authorization	13
Format	13
Parameters	13
Return and Reason Codes	13
Usage Notes	14
Related Services	14
ck_IPC_access (IRRSKI00): Check IPC Access	14
Function	14
Requirements	14

RACF Authorization	14
Format	15
Parameters	15
Return and Reason Codes	16
Usage Notes	16
Related Services	16
ck_owner_two_files (IRRS200): Check Owner of Two Files	16
Function	16
Requirements	16
RACF Authorization	16
Format	17
Parameters	17
Return and Reason Codes	18
Usage Notes	18
Related Services	18
ck_priv (IRRSKP00): Check Privilege	18
Function	18
Requirements	18
RACF Authorization	19
Format	19
Parameters	19
Return and Reason Codes	20
Usage Notes	20
Related Services	20
ck_process_owner (IRRSKO00): Check Process Owner	20
Function	20
Requirements	20
RACF Authorization	21
Format	21
Parameters	21
Return and Reason Codes	22
Usage Notes	22
Related Services	22
clear_setid (IRRS000): Clear Set ID	22
Function	22
Requirements	22
RACF Authorization	23
Format	23
Parameters	23
Return and Reason Codes	24
Usage Notes	24
Related services	24
deleteUSP (IRRS000): Delete USP	24
Function	24
Requirements	24
RACF Authorization	24
Format	25
Parameters	25
Return and Reason Codes	25
Usage Notes	25
Related Services	25
getGMAP (IRRS000): Get GID-to-Group-Name Mapping	25
Function	25

Requirements	26	Return and reason codes	54
RACF Authorization	26	Usage Notes	54
Format	26	Related services	55
Parameters	26	make_root_FSP (IRRSMR00): Make Root IFSP	55
Return and Reason Codes	27	Function	55
Usage Notes	27	Requirements	55
Related Services	27	RACF Authorization	55
get_uid_gid_supgrps (IRRSGE00): Get UIDs, GIDs, and Supplemental Groups	28	Format	55
Function	28	Parameters	55
Requirements	28	Return and Reason Codes	56
RACF Authorization	28	Usage Notes	56
Format	28	Related Services	57
Parameters	29	query_file_security_options (IRRSQF00): Query File Security Options	57
Return and Reason Codes	30	Function	57
Usage Notes	30	Requirements	57
Related Services	30	RACF Authorization	57
getUMAP (IRRSUM00): Get UID-to-User-ID Mapping	30	Format	58
Function	30	Parameters	58
Requirements	30	Return and Reason Codes	59
RACF Authorization	31	Usage Note	59
Format	31	Related Services	59
Parameters	31	query_system_security_options (IRRSQS00): Query System Security Options	59
Return and Reason Codes	32	Function	59
Usage Notes	32	Requirements	59
Related Services	32	RACF Authorization	59
initACEE (IRRSIA00): Initialize ACEE	32	Format	60
Function	32	Parameters	60
Requirements	33	Return and Reason Codes	60
Linkage Conventions	33	Usage Note	61
RACF Authorization	33	Related Services	61
Format	34	R_admin (IRRSEQ00): RACF Administration API	61
Parameters	34	Function	61
Return and Reason Codes	39	Requirements	61
Usage Notes	42	RACF authorization	61
Related Services	46	Format	62
initUSP (IRRSIU00): Initialize USP	46	Parameters	62
Function	46	Return and reason codes	65
Requirements	47	Usage notes	66
RACF Authorization	47	Related services	68
Format	47	Parameter list formats	68
Parameters	47	R_audit (IRRSAU00): Provide an audit interface	99
Return and Reason Codes	48	Function	99
Usage Notes	48	Requirements	99
Related Services	49	RACF authorization	99
makeFSP (IRRSMF00): Make IFSP	49	Format	100
Function	49	Parameters	100
Requirements	49	Return and reason codes	101
RACF Authorization	49	Usage notes	101
Format	50	Related services	101
Parameters	50	R_cacheserv (IRRSCH00): Cache Services	101
Return and Reason Codes	50	Function	101
Usage Notes	51	Requirements	101
Related Services	52	Linkage conventions	102
makeISP (IRRSMI00): Make IISP	52	RACF authorization	102
Function	52	Format	102
Requirements	52	Parameters	102
RACF Authorization	53	Return and reason codes	105
Format	53	Parameter usage	106
Parameters	53	Usage notes	107

Related services	108	Format	132
R_chaudit (IRRSCA00): Change Audit Options	108	Parameters	132
Function	108	Return and Reason Codes	132
Requirements	108	Usage Notes.	133
RACF authorization	108	Related Services	134
Format	108	R_dceruid (IRRSUD00): Determine the ID of a	
Parameters	108	Client	134
Return and reason codes	109	Function	134
Usage notes	110	Requirements	134
Related Services	110	RACF Authorization	135
R_chmod (IRRSCF00): Change File Mode	110	Format	135
Function	110	Parameters	135
Requirements	110	Return and Reason Codes	136
RACF Authorization	111	Usage Notes.	137
Format	111	Related Services	137
Parameters	111	R_exec (IRRSEX00): Set Effective and Saved	
Return and Reason Codes	112	UIDs/GIDs	137
Usage Notes.	112	Function	137
Related Services	112	Requirements	138
R_chown (IRRSO00): Change Owner and Group	112	RACF Authorization	138
Function	112	Format	138
Requirements	112	Parameters	138
RACF Authorization	113	Return and Reason Codes	139
Format	114	Usage Notes.	139
Parameters	114	Related Services	139
Return and Reason Codes	115	R_fork (IRRSFK00): Fork a Process	139
Usage Notes.	115	Function	139
Related Services	115	Requirements	140
R_datalib (IRRSDL00): OCSF Data Library.	115	RACF Authorization	140
Function	115	Format	140
Requirements	115	Parameters	140
Linkage Conventions	116	Return and Reason Codes	141
RACF Authorization	116	Usage Notes.	141
Format	116	Related Services	142
Parameters	116	R_getgroups (IRRSYG00): Get/Set Supplemental	
Return and Reason Codes	118	Groups	142
Usage Notes.	121	Function	142
Related Services	123	Requirements	142
R_dceauth (IRRSDA00): Check a User's Authority	123	RACF Authorization	143
Function	123	Format	143
Requirements	123	Parameters	143
RACF Authorization	123	Return and Reason Codes	144
Format	124	Usage Note	144
Parameters	124	Related Services	144
Return and Reason Codes	125	R_getgroupsbyname (IRRSUG00): Get Groups by	
Usage Notes.	126	Name	144
Related Services	126	Function	144
R_dceinfo (IRRSDI00): Retrieve or Set User Fields	126	Requirements	145
Function	126	RACF Authorization	145
Requirements	127	Format	145
RACF Authorization	127	Parameters	145
Format	127	Return and Reason Codes	146
Parameters	128	Usage Notes.	146
Return and Reason Codes	129	Related Services	146
Usage Notes.	130	R_IPC_ctl (IRRSI00): Perform IPC Control	147
Related Services	130	Function	147
R_dcekey (IRRSK00): Retrieve or Set a non-RACF		Requirements	147
Password.	131	RACF Authorization	147
Function	131	Format	148
Requirements	131	Parameters	148
RACF Authorization	131	Return and Reason Codes	149

Usage Notes	149	Related Services	199
Related Services	150	R_setfacl (IRRSL00):Unix Access Control Lists	199
R_kerbinfo (IRRSMK00): Retrieve or Set Security		Function	199
Server Network Authentication Service Fields	150	Requirements	199
Function	150	RACF authorization	200
Requirements	150	Format	200
Linkage Conventions	151	Parameters	200
Format	151	Return and reason codes	201
Parameters	151	Usage notes	201
Return and Reason Codes	154	Related services	202
Usage Notes	155	R_setgid (IRRSSG00): Set Group Name	202
Parameter Usage	155	Function	202
Related Services	155	Requirements	202
R_PKIServ (IRRSPX00): Request Public Key		RACF Authorization	203
Infrastructure (PKI) Services	155	Format	203
Function	155	Parameters	203
Requirements	156	Return and Reason Codes	204
RACF Authorization	156	Usage Notes	204
Format	157	Related Services	204
Parameters	158	R_setuid (IRRSSU00): Set z/OS UNIX user	
Return and Reason Codes	179	identifier (UID)	204
Usage Notes	183	Function	204
R_proxyserv (IRRSPY00): LDAP interface	187	Requirements	204
Function	187	RACF Authorization	205
Requirements	187	Format	205
Linkage Conventions	187	Parameters	205
RACF Authorization	187	Return and Reason Codes	205
Format	187	Usage Notes	206
Parameters	188	Related Services	206
Return and Reason Codes	190	R_ticketserv (IRRSPK00): Parse or Extract	206
Parameter Usage	192	Function	206
Usage Notes	193	Requirements	206
Related Services	193	Linkage Conventions	206
R_ptrace (IRRSP00): Ptrace Authority Check	194	RACF Authorization	207
Function	194	Format	207
Requirements	194	Parameters	207
RACF Authorization	194	Return and Reason Codes	208
Format	194	Usage Notes	209
Parameters	194	Parameter Usage	210
Return and Reason Codes	195	Related Services	210
Usage Notes	195	R_umask (IRRSM00): Set File Mode Creation	
Related Services	195	Mask	210
R_setegid (IRRSEG00): Set Effective GID, Set All		Function	210
GIDs	195	Requirements	210
Function	195	RACF Authorization	211
Requirements	196	Format	211
RACF Authorization	196	Parameters	211
Format	196	Return and Reason Codes	212
Parameters	196	Usage Note	212
Return and Reason Codes	197	Related services	212
Usage Notes	197	R_usermap (IRRSIM00): Map application user	212
Related Services	197	Function	212
R_seteuid (IRRSEU00): Set Effective UID, Set All		Requirements	212
UIDs	197	Linkage Conventions	213
Function	197	RACF Authorization	213
Requirements	197	Format	213
RACF Authorization	198	Parameters	213
Format	198	Return and Reason Codes	214
Parameters	198	Parameter Usage	215
Return and Reason Codes	199	Usage Notes	215
Usage Notes	199	Related Services	217

Chapter 3. IRRSXT00 Installation Exit	219
Function	219
Requirements	219
Interface Registers	220
Input	220
Output	220
Usage Notes.	221
Appendix. Accessibility	223

Using assistive technologies	223
Keyboard navigation of the user interface	223

Notices	225
Programming Interface Information	226
Trademarks	226

Index	229
------------------------	------------

Tables

1. Intended Use of RACF Callable Services	7	49. BASE Segment Fields	84
2. UNIXPRIV class resource names used in ck_access	10	50. DLFDATA Segment Fields	86
3. UNIXPRIV class resource names used in ck_owner_two_files	17	51. SESSION Segment Fields	86
4. UNIXPRIV class resource names used in ck_priv	19	52. SSIGNON Segment Fields.	87
5. UNIXPRIV class resource names used in ck_process_owner	21	53. STDATA Segment Fields	87
6. Values Allowed for Attributes Parameter	37	54. SVFMR Segment Fields	87
7. ENVR Data Structure	38	55. TME Segment Fields	88
8. ENVR_out Storage Area Processing	38	56. KERB Segment Fields	88
9. X500 Name Pair Data Structure	39	57. PROXY Segment Fields	89
10. Criteria Value Data Structure.	39	58. EIM Segment Fields.	89
11. initACEE Create Return Codes	39	59. BASE Segment Fields	89
12. initACEE Delete Return Codes	40	60. DFP Segment Fields.	91
13. initACEE Purge Return Codes	40	61. TME Segment Fields	92
14. initACEE Register and Deregister Return Codes	40	62. Base Segment Fields	92
15. initACEE Query Return Codes	41	63. Parameter List Mapping for SETROPTS Administration	92
16. Parameter Usage.	41	64. Segment Entry Fields	93
17. Function Code Values in Mapping Macro IRRPCOMP	63	65. Field Entry Format	93
18. Parameter List Mappings for Function_Code Values	64	66. BASE Segment Field Names	94
19. Mapping of Output Message Block.	64	67. Output Message Block	98
20. Format of Each Message Entry	65	68. UNIXPRIV class resource names used in R_chown	113
21. Return and Reason Codes.	65	69. IRRSDL00 Return Codes	118
22. Parameter List Format for Running a Command	68	70. DataGetFirst and DataGetNext Return Codes	121
23. Parameter List Format for User Administration	68	71. DataAbortQuery Return Codes	122
24. Segment Entry Mapping	68	72. CheckStatus Return Codes	123
25. Field Entry Mapping	69	73. UNIXPRIV class resource names used in R_IPC_ctl	148
26. BASE Segment Fields	70	74. ENCTYPE field value.	154
27. OMVS Segment Fields	71	75. Parameter Usage	155
28. TSO Segment Fields.	71	76. Function_parmlist for GENCERT	159
29. CICS Segment Fields	72	77. CertPlist for GENCERT	160
30. NetView Segment Fields	73	78. Function_parmlist for EXPORT	162
31. DCE Segment Fields	74	79. Function_parmlist for QUERYREQS	162
32. DFP Segment Fields.	74	80. ResultsList for QUERYREQS	164
33. Language Segment Fields	74	81. Function_parmlist for REQDETAILS	165
34. OPERPARM Segment Fields	74	82. SumList for REQDETAILS	166
35. OVM Segment Fields	75	83. CertPlist for REQDETAILS	167
36. WORKATTR Segment Fields.	76	84. Function_parmlist for MODIFYREQS	168
37. LNOTES Segment Fields	76	85. CertPlist for MODIFYREQS	169
38. NDS Segment Fields	76	86. Function_parmlist for QUERYCERTS	170
39. KERB Segment Fields	77	87. ResultsList for QUERYCERTS	171
40. PROXY Segment Fields	77	88. Function_parmlist for CERTDETAILS	173
41. EIM Segment Fields.	77	89. SumList for CERTDETAILS	174
42. BASE Segment Fields	79	90. CertPlist for CERTDETAILS.	174
43. DFP Segment Fields.	80	91. Function_parmlist for MODIFYCERTS	175
44. OMVS Segment Fields	80	92. Function_parmlist for VERIFY	175
45. OVM Segment Fields	80	93. SumList for VERIFY	176
46. TME Segment Fields	80	94. CertPlist for VERIFY	177
47. Base Segment Fields	82	95. Function_parmlist for REVOKE	177
48. Resource Related Field Definitions	84	96. Function_parmlist for GENRENEW and REQRENEW	178
		97. CertPlist for GENRENEW and REQRENEW	178
		98. Return and Reason Codes	179
		99. Result_entries output area	189

100. UNIXPRIV class resource names used in
R_ptrace 194

101. Parameter Usage 215

About this document

This document supports z/OS (5694-A01) and z/OS.e (5655-G52) and contains information about the Resource Access Control Facility (RACF), which is part of the Security Server. The Security Server has these components:

- RACF
- DCE Security Server
- OS/390 Firewall Technologies
- Lightweight Directory Access Protocol (LDAP) Server, which includes client and server function
- Open Cryptographic Enhanced Plug-ins (OCEP)
- Security Server Network Authentication Service
- PKI Services

For information about these components, see the documents related to them.

This document documents callable services provided by RACF. It contains:

- Information on using the callable services
- A description of each callable service
- Descriptions of the data areas used by the callable services
- A description of an installation exit that can be used in conjunction with the callable services

Intended audience

This document is intended for system programmers who are familiar with RACF concepts and terminology. They should also be familiar with MVS systems and with z/OS UNIX.

IBM systems center publications

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

G320-9279	<i>Systems Security Publications Bibliography</i>
GG22-9396	<i>Tutorial: Options for Tuning RACF</i>
GG24-3378	<i>DFSMS and RACF Usage Considerations</i>
GG24-3451	<i>Introduction to System and Network Security: Considerations, Options, and Techniques</i>
GG24-3524	<i>Network Security Involving the NetView® Family of Products</i>
GG24-3970	<i>Elements of Security: RACF Overview - Student Notes</i>
GG24-3971	<i>Elements of Security: RACF Installation - Student Notes</i>
GG24-3972	<i>Elements of Security: RACF Advanced Topics - Student Notes</i>
GG24-3984	<i>RACF Macros and Exit Coding</i>
GG24-4282	<i>Secured Single Signon in a Client/Server Environment</i>

GG24-4453	<i>Enhanced Auditing Using the RACF SMF Data Unload Utility</i>
GG26-2005	<i>RACF Support for Open Systems Technical Presentation Guide</i>
GC28-1210	<i>System/390[®] MVS[™] Sysplex Hardware and Software Migration</i>
SG24-4704	<i>OS/390[®] Security Services and RACF-DCE Interoperation</i>
SG24-4820	<i>OS/390 Security Server Audit Tool and Report Application</i>
SG24-5158	<i>Ready for e-business: OS/390 Security Server Enhancements</i>
SG24-5339	<i>The OS/390 Security Server Meets Tivoli[®]: Managing RACF with Tivoli Security Products</i>

Other sources of information

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

IBM discussion areas

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

Internet sources

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

- **Online library**

To view and print online versions of the z/OS[™] publications, use this address:
<http://www.ibm.com/servers/eserver/zseries/zos/bkserv/>

- **Redbooks[™]**

The documents known as Redbooks that are produced by the International Technical Support Organization (ITSO) are available at the following address:
<http://www.ibm.com/redbooks/>

- **Enterprise systems security**

For more information about security on the S/390[®] platform, OS/390, and z/OS, including the elements that comprise the Security Server, use this address:
<http://www.ibm.com/servers/eserver/zseries/zos/security/>

- **RACF home page**

You can visit the RACF home page on the World Wide Web using this address:
<http://www.ibm.com/servers/eserver/zseries/zos/racf/>

- **RACF-L discussion list**

Customers and IBM participants may also discuss RACF on the RACF-L discussion list. RACF-L is not operated or sponsored by IBM; it is run by the University of Georgia.

To subscribe to the RACF-L discussion and receive postings, send a note to:
listserv@listserv.uga.edu

Include the following line in the body of the note, substituting your first name and last name as indicated:

```
subscribe racf-l first_name last_name
```

To post a question or response to RACF-L, send a note, including an appropriate Subject: line, to:

racf-l@listserv.uga.edu

- **Sample code**

You can get sample code, internally-developed tools, and exits to help you use RACF. This code works in our environment, at the time we make it available, but is not officially supported. Each tool or sample has a README file that describes the tool or sample and any restrictions on its use.

To access this code from a Web browser, go to the RACF home page and select the “Downloads” topic from the navigation bar, or go to `ftp://ftp.software.ibm.com/eserver/zseries/zos/racf/`.

The code is also available from `ftp.software.ibm.com` through anonymous FTP. To get access:

1. Log in as user **anonymous**.
2. Change the directory, as follows, to find the subdirectories that contain the sample code or tool you want to download:

```
cd eserver/zseries/zos/racf/
```

An announcement will be posted on the RACF-L discussion list and on newsgroup *ibm.servers.mvs.racf* whenever something is added.

Note: Some Web browsers and some FTP clients (especially those using a graphical interface) might have problems using `ftp.software.ibm.com` because of inconsistencies in the way they implement the FTP protocols. If you have problems, you can try the following:

- Try to get access by using a Web browser and the links from the RACF home page.
- Use a different FTP client. If necessary, use a client that is based on command line interfaces instead of graphical interfaces.
- If your FTP client has configuration parameters for the type of remote system, configure it as UNIX[®] instead of MVS.

Restrictions

Because the sample code and tools are not officially supported,

- There are no guaranteed enhancements.
- No APARs can be accepted.

To request copies of IBM publications

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

There is also a toll-free customer support number (1-800-879-2755) available Monday through Friday from 6:30 a.m. through 5:00 p.m. Mountain Time. You can use this number to:

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

Where to find more information

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

Softcopy publications

The RACF library is available on the following CD-ROMs. The CD-ROM online library collections include Library Reader™, which is a program that enables you to view the softcopy documents.

SK3T-4269 *z/OS Version 1 Release 4 Collection*

This collection contains the set of unlicensed documents for the current release of z/OS in both BookManager® and Portable Document Format (PDF) files. You can view or print the PDF files with the Adobe Acrobat reader.

SK3T-4272 *z/OS Security Server RACF Collection*

This softcopy collection kit contains the Security Server library for z/OS in both BookManager and Portable Document Format (PDF) files. You can view or print the PDF files with the Adobe Acrobat reader.

SK2T-2180 *Online Library OS/390 Security Server RACF Information Package*

This softcopy collection contains the Security Server library for OS/390. It also contains the RACF/MVS Version 2 product libraries, the RACF/VM 1.10 product library, product documents from the OS/390 and VM collections, International Technical Support Organization (ITSO) documents (known as Redbooks), and Washington System Center (WSC) documents (known as orange books) that contain information related to RACF. The collection does not contain any licensed publications. By using this CD-ROM, you have access to RACF-related information from IBM products such as OS/390, VM/ESA®, CICS®, and NetView.

SK3T-7876 *IBM eServer zSeries™ Redbooks Collection*

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

SK2T-2177 *IBM Redbooks S/390 Collection*

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

RACF courses

The following RACF classroom courses are available:

ES840 *Implementing RACF Security for CICS/ESA® and CICS/TS*

H3917 *Basics of OS/390 Security Server RACF Administration*

H3927 *Effective RACF Administration*

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

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

Using LookAt to look up message explanations

LookAt is an online facility that allows you to look up explanations for most messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can access LookAt from the Internet at:

<http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/>

or from anywhere in z/OS where you can access a TSO/E command line (for example, TSO/E prompt, ISPF, z/OS UNIX System Services running OMVS). You can also download code from the *z/OS Collection* (SK3T-4269) and the LookAt Web site that will allow you to access LookAt from a handheld computer (Palm Pilot VIIx suggested).

To use LookAt as a TSO/E command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO/E from a disk on your *z/OS Collection* (SK3T-4269) or from the **News** section on the LookAt Web site.

Some messages have information in more than one document. For those messages, LookAt displays a list of documents in which the message appears.

Accessing z/OS licensed documents on the Internet

z/OS licensed documentation is available on the Internet in PDF format at the IBM Resource Link™ Web site at:

<http://www.ibm.com/servers/resourceLink>

Licensed documents are available only to customers with a z/OS license. Access to these documents requires an IBM Resource Link user ID and password, and a key code. With your z/OS order you received a Memo to Licensees, (GI10-0671), that includes this key code.¹

To obtain your IBM Resource Link user ID and password, log on to:

<http://www.ibm.com/servers/resourceLink>

To register for access to the z/OS licensed documents:

1. Sign in to Resource Link using your Resource Link user ID and password.
2. Select **User Profiles** located on the left-hand navigation bar.

Note: You cannot access the z/OS licensed documents unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

Printed licensed documents are not available from IBM.

1. z/OS.e™ customers received a Memo to Licensees, (GI10-0684) that includes this key code.

You can use the PDF format on either **z/OS Licensed Product Library CD-ROM** or IBM Resource Link to print licensed documents.

Summary of changes

Summary of changes for SA22-7691-03 z/OS Version 1 Release 4

This book contains information previously presented in z/OS SecureWay Security Server RACF Callable Services, SA22-7691-02, which supports z/OS Version 1 Release 3.

The following summarizes the changes to that information.

New information

- Information is added to indicate this document supports z/OS.e.
- Callable service, R_admin (IRRSEQ00), is changed to support new fields for EIM for the User and General Resource fields. New information has also been added to support new SHARED and AUTOUID/AUTOGID keywords.
- Callable service, R_data lib (IRRSDL00), is changed to support a return to PCICC key types.
- Callable service, R_dcekey (IRRSDK00), is changed to support a LDAP bind password.
- Callable service, makeFSP (IRRSMF00), is changed to support the FILE.GROUPOWNER.SETGID profile in the UNIXPRIV class.
- Callable service, R_PKIServ (IRRSPX00), is changed to support MAIL, STREET, and POSTALCODE distinguished name qualifiers.
- New information has been added to callable service, ck_access (IRRSKA00). RACF Authorization has added information about UNIXPRIV.

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

Starting with z/OS V1R2, you may notice changes in the style and structure of some content in this book—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our books.

Summary of changes for SA22-7691-02 z/OS Version 1 Release 3

This book contains information previously presented in z/OS SecureWay Security Server RACF Callable Services, SA22-7691-01, which supports z/OS Version 1 Release 2.

The following summarizes the changes to that information.

New information

- Callable service, R_cacheserv (IRRSCH00), has been added.
- Callable service, R_proxyserv (IRRSPY00), has been added.

- Callable service, R_setfac1 (IRRSC00), has been added.
- Callable service, R_admin (IRRSEQ00), is changed to support a new field name, PROXY, for the User and General Resource functions to the BASE Segment Fields table.
- Callable service, ck_access (IRRSKA00), is changed to document the use of SUPERUSER.FILESYS in support of ACLs.
- Callable service, query_file_security_options (IRRSQF00), has a new option code value of 2 added, defined in Parameters.
- Callable service, R_PKIServ (IRRSPX00), has added SAF trace information to the Function section.
- Callable service, makeFSP (IRRSMF00), is changed to support new Return and Reason Codes and new Usage Notes in support of ACLs.
- An appendix with z/OS product accessibility information was added.

Deleted information

- Note that the glossary has been removed from this book. You can now find the glossary in the *z/OS Security Server RACF Security Administrator's Guide*

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

Summary of changes for SA22-7691-01 z/OS Version 1 Release 2

This book contains information previously presented in SA22-7691-00, which supports z/OS Version 1 Release 1.

The following summarizes the changes to that information.

New information

- Callable service, R_admin (IRRSEQ00), is changed to support a new field name. ENCRYPT, for the user, resource, and system options functions KERB Segment Fields Table.
- Callable service, R_admin (IRRSEQ00), is changed to support a new field name, UNIVERSL, to group functions BASE Segment Table.
- New information has been added to callable service, R_chmod (IRRSCF00). RACF Authorization has added information about the UNIXPRIV class.
- Callable service, R_kerbinf0 (IRRSMK00), is changed to support the new format for value fields CURKEY and PREVKEY.

This book contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

Chapter 1. Using the RACF Callable Services

The RACF security functions provided for use by z/OS UNIX and other products integrated with it are called as callable services. Normal installation applications using the services or functions of z/OS UNIX cannot call the RACF callable services directly. They must use the z/OS UNIX callable services instead.

Linkage Conventions for the Callable Services

The linkage is created as follows:

- For non-IBM modules, or modules written in languages other than PL/X, the CALL statement must generate a V-type constant (VCON) with the module name of a stub routine for the requested service. The module names are defined as part of the callable services interface described in Chapter 2, “Callable Services Descriptions” on page 7. The VCON can be resolved by link-editing the control section (CSECT) with the stub routines provided as part of MVS’s system authorization facility (SAF). There is a stub for each service.
- The linkage loads a function code indicating the service requested and calls the callable services router. The function codes that can be used are described in *z/OS Security Server RACF Data Areas*.
- The callable services router calls an installation exit (IRRSXT00) and then calls the RACF router.
- The RACF router invokes the requested service routine based on the function code.
- The service routine provides the requested function and returns to the SAF callable services router.
- The SAF callable services router calls the installation exit IRRSXT00 a second time, sets the SAF return code, and returns to the caller.

Work Area (WORK)

When a module calls a RACF callable service, it must provide the address of a work area. The work area is a 1024-byte structure that is used by SAF, RACF, and the SAF exit routine IRRSXT00. IRRSXT00 can use the first 152 bytes of the area. The first 16 bytes are preserved from the pre-RACF exit invocation to the post-RACF exit invocation and can be used to pass parameters.

For the mapping of the work area, see *z/OS Security Server RACF Data Areas*. For information on IRRSXT00, see Chapter 3, “IRRSXT00 Installation Exit” on page 219.

File Security Packet (IFSP)

Security-relevant data for files in the z/OS UNIX file system is kept in a file security packet (IFSP) structure owned by RACF. The IFSP is stored in the file system as part of the attributes associated with a file. When a file is created, the IFSP is created by the **makeFSP** or the **make_root_FSP** callable service. The **makeFSP** service returns an IFSP to the file system, which writes it with other attributes of the file. On subsequent accesses to the file, the file system reads the IFSP and passes it to other callable services. The file system deletes the IFSP when the file is deleted.

The IFSP is a fixed-size 64-byte area. It is written to storage as part of the PFAR for the file and its size cannot be changed.

The file system manages the storage for the IFSP. The **makeFSP** service fills in the data, and other callable services use or modify the data in the area provided by the file system.

The IFSP data can be examined by users other than the security product. The IFSP is mapped by macro IRRPIFSP. Others should not use this mapping to create or directly modify the IFSP, and should not make their own security or audit decisions based on the contents of the IFSP.

The IFSP contains the following data:

- Control block ID
- Version number
- z/OS UNIX user identifier (UID) of the owner of the file
- z/OS UNIX group identifier (GID) of the group owner of the file
- Mode bits:
 - Owner permission bits
 - Group permission bits
 - Other permission bits
 - S_ISUID, S_ISGID, and S_ISVTX bits
- User audit options for the file
- Auditor audit options for the file

For the mapping of the file security packet, see *z/OS Security Server RACF Data Areas*.

Security Credentials (CRED)

The security credentials (CRED) structure is used in the z/OS UNIX file system to pass data from the logical file system (LFS) through the physical file system (PFS) to the RACF callable services.

The CRED is built by the LFS, and is created for each system call entry to the LFS. The CRED is used for all `vm_ops` called (and most RACF callable service calls by the PFS) for the system call. The CRED is not kept across multiple LFS system calls.

The CRED contains:

- **User information:** a user type field that indicates whether the caller is a standard z/OS UNIX process known to RACF, or a system function that is not a process.

Functions that accept a system caller process the request as if the caller is a superuser. If an audit record is written, the user z/OS UNIX user identifier (UID) and z/OS UNIX group identifier (GID) values in the record are set to -1.

- **Audit data:** data known by the LFS that needs to be passed through the PFS to the RACF callable services for auditing. This data is:
 - **Audit function code:** a code that identifies the system call being processed. The audit function codes are described in *z/OS Security Server RACF Data Areas*.
 - **Name flag:** a flag used on path resolution calls to `ck_access` to indicate whether the first or second file name is being checked.

- **Requested path name:** the path name the user passed on the system call. For link, vlink, rename, and vrename, this is the old path name. When the caller of lookup is getcwd, ioctl, or ttyname, this field is not filled in.
- **File name** the part of the requested path name currently being checked. This may be part of the path name or may be part of a symbolic link encountered when resolving the path name. The first directory checked in a path name resolution is either the root directory (**/ROOT**) or the current working directory (**/CWD**). The names **/ROOT** and **/CWD**—the only file names that contain a slash (/)—are provided to indicate these directories in the audit record. This field is included only in audit records produced by **ck_access**. This field contains the file name of:
 - The directory being checked on calls from lookup.
When the caller of lookup is getcwd, ioctl, or ttyname, this field is not filled in.
 - The parent directory of the object identified by the pathname for calls for mkdir, mknod, vcreate, open(new file), rename, vrename, rmdir, symlink, vsymlink, unlink, and vremove.
 - The object identified by the path name for calls for open(old file), opendir, link, vlink, and utime.
 - **Second path name:** for rename, vrename, link, and vlink, this is the new path name passed on the system call. For symlink and vsymlink, this is the content of the symlink. For mount and unmount, this is the data set name of the HFS data set being mounted or dismounted.
 - **Second file name:** this is the same as the file name above, except that it is for the second part of the path name being checked. This field contains the file name of:
 - The directory being checked on calls from lookup
 - The parent directory of the object identified by the new pathname for calls for link, vlink, rename, and vrename.

The CRED structure is mapped by the IRRPCRED mapping macro.

For the mapping of the CRED, see *z/OS Security Server RACF Data Areas*.

File identifiers

————— **Programming Interface** —————

Part of the audit data for file access is a file identifier. The file identifier is a 16-byte token that uniquely identifies a file while it is mounted on the system.

————— **End of Programming Interface** —————

————— **Programming Interface** —————

If the file system is unmounted and remounted, that file identifier may change. A change in file identifiers can be detected in the audit trail by matching the mount audit records with the same file system name and comparing the file identifiers for the root directory.

————— **End of Programming Interface** —————

File Type and File Mode Values

Programming Interface

A mode value is input to z/OS UNIX `chmod`, `open`, `creat`, `mkdir`, and `umask`, and output by z/OS UNIX `stat` and `fstat`. The mode value is defined as a `mode_t` data type and consists of a one-byte file type and three bytes for the file modes. The file mode specifies the permission bits and the `S_ISUID`, `S_ISGID`, and `S_ISVTX` bits for a file.

End of Programming Interface

Programming Interface

The z/OS UNIX macro `BPXYMODE` defines the `mode_t` values as:

- Bits 0–7: file type, mapped by z/OS UNIX macro `BPXYFTYP`
- Bits 8–13: reserved
- Bits 14–31: available to the security product:
 - Bits 14–19: reserved
 - Bit 20: `S_ISUID` (set user ID on execution)
 - Bit 21: `S_ISGID` (set group name on execution)
 - Bit 22: `S_ISVTX` (keep loaded executable in storage)
 - Bits 23–25: `S_IRWXU` (owner class mask)
 - Bit 23: `S_IRUSR` (read permission)
 - Bit 24: `S_IWUSR` (write permission)
 - Bit 25: `S_IXUSR` (search (if directory) or execute (otherwise) permission)
 - Bits 26–28: `S_IRWXG` (group class mask)
 - Bit 26: `S_IRGRP` (read permission)
 - Bit 27: `S_IWGRP` (write permission)
 - Bit 28: `S_IXGRP` (search (if directory) or execute (otherwise) permission)
 - Bits 29–31: `S_IRWXO` (other class mask)
 - Bit 29: `S_IROTH` (read permission)
 - Bit 30: `S_IWOTH` (write permission)
 - Bit 31: `S_IXOTH` (search (if directory) or execute (otherwise) permission)

End of Programming Interface

Programming Interface

The system call services pass the mode parameter from the caller of the system call to the RACF callable service or from the RACF callable service to the caller of the system call. The system call service can change the file type but does not change the file mode bits.

End of Programming Interface

Programming Interface

Some RACF callable services test the file type to determine if the file is a directory. The `makeFSP` service sets the file type to “directory” if the file is a directory and

sets it to zero otherwise.

End of Programming Interface

IPC Security Packet (IISP)

Interprocess communication (IPC) requires RACF to do authorization and permission checking. IPC facilities of the z/OS UNIX system allow two or more distinct processes to communicate with each other. RACF protects this environment so that only those processes with the correct authority can communicate.

Interprocess communication consists of message queueing, semaphores, and shared memory segments used by application programs. Each function requires a security action by z/OS UNIX, which RACF performs to allow a secure environment to exist.

The IPC security packet (IISP) contains data needed to make security decisions. It is built when a new ID for an IPC key is created and is saved in memory by the kernel. The IISP is used in place of a profile in the RACF database to contain information about the IPC key's owner and access rights.

The **makeISP** service initializes the IPC security packet (IISP) for a new IPC key with the creator's user and group identifiers (UID and GID), the owner's UID and GID, the mode bits, the IPC key, and the IPC ID.

The **ck_IPC_access** service determines whether the current process has the requested access to an IPC key. The IISP of the key is passed with this request. The **ck_IPC_access** service is called separately for each IPC key.

For the z/OS UNIX IPC_SET command, the **R_IPC_ctl** service modifies the owner's UID, owner's GID, and mode bits in the IISP for the IPC key if the authority is correct. For the z/OS UNIX IPC_RMID command, the **R_IPC_ctl** service checks the authority of the current process to determine whether the resource can be removed.

The IISP consists of two parts, the root and the extension. The root is mapped by macro IRRPIISP. The root contains a pointer to the extension, which is mapped by the z/OS UNIX mapping macro BPXYIPCP. Other products can read the IISP for reporting purposes using the IRRPIISP and BPXYIPCP mapping macros.

The IISP root contains the following data:

- Control block ID
- Version number
- ALET of the IPCP
- Address of the IPCP (mapped by z/OS UNIX macro BPXYIPCP)
- IPC key
- IPC ID

For the mapping of the IPC security packet, see *z/OS Security Server RACF Data Areas*.

Interprocess Communications Permission (BPXYIPCP)

```
BPXYIPCP ,
** BPXYIPCP: Interprocess Communications Permission
** Used By: MCT, MGT, SCT, SGT, QCT, QGT
IPC_PERM          DSECT ,      Interprocess Communications
IPC_UID           DS    F      Owner's effective user ID
IPC_GID           DS    F      Owner's effective group name
IPC_CUID          DS    F      Creator's effective user ID
IPC_CGID          DS    F      Creator's effective group name
IPC_MODE          DS    XL4     Mode, mapped by BPXYMODE
IPC#LENGTH       EQU  *-IPC_PERM Length of Interprocess Control block
* Key:
IPC_PRIVATE       EQU    0      Private key.
* Mode bits:
IPC_CREAT         EQU    1      Create entry if key does not exist.
IPC_EXCL          EQU    2      Fail if key exists.
* Flag bits - semop, msgrcv, msgsnd:
IPC_NOWAIT        EQU    1      Error if request must wait.
* Control Command:
IPC_RMID          EQU    1      Remove identifier.
IPC_SET           EQU    2      Set options.
IPC_STAT          EQU    3      Access status.
* CONSTANTS WHICH MAP OVER BYTE S_TYPE, SEE BPXYMODE
** BPXYIPCP End
```

IPC Security Credentials (CREI)

The IPC security credentials (CREI) structure is used in the z/OS UNIX IPC system to pass data from the kernel to RACF.

The CREI is built by the kernel, and is created for each system call entry to RACF.

The CREI contains:

- **User information:** a user type field that indicates whether the caller is a standard z/OS UNIX process known to RACF, or a system function that is not a process.
Functions that accept a system caller process the request as if the caller is a superuser. If an audit record is written, the user z/OS UNIX user identifier (UID) and z/OS UNIX group identifier (GID) values in the record are set to -1.
- **Audit data:** data known by the kernel that needs to be passed through the IPC system to the RACF callable services for auditing. This data includes an **audit function code**, which identifies the system call being processed. The audit function codes are described in *z/OS Security Server RACF Data Areas*.
- **IPC key:** the key of the IPC service that is being checked.
- **IPC identifier:** the identifier of the IPC service that is being checked.

The CREI structure is mapped by the IRRPCREI mapping macro.

For the mapping of the CREI, see *z/OS Security Server RACF Data Areas*.

Chapter 2. Callable Services Descriptions

This chapter describes the RACF callable services. The services appear in alphabetic order. Table 1 lists each callable service's intended users.

Table 1. Intended Use of RACF Callable Services

Callable Service	For Use By
"ck_access (IRRSKA00): Check Access" on page 9	z/OS UNIX file system or z/OS UNIX servers
"ck_file_owner (IRRSKF00): Check File Owner" on page 12	z/OS UNIX file system or z/OS UNIX servers
"ck_IPC_access (IRRSKI00): Check IPC Access" on page 14	MVS BCP or z/OS UNIX task level processes
"ck_owner_two_files (IRRSK200): Check Owner of Two Files" on page 16	z/OS UNIX file system and z/OS UNIX servers.
"ck_priv (IRRSKP00): Check Privilege" on page 18	z/OS UNIX file system, MVS BCP, or z/OS UNIX servers
"ck_process_owner (IRRSKO00): Check Process Owner" on page 20	MVS BCP or z/OS UNIX task level processes
"clear_setid (IRRSKS00): Clear Set ID" on page 22	z/OS UNIX file system or z/OS UNIX servers
"deleteUSP (IRRSU00): Delete USP" on page 24	MVS BCP or z/OS UNIX servers
"getGMAP (IRRSKM00): Get GID-to-Group-Name Mapping" on page 25	MVS BCP
"get_uid_gid_supgrps (IRRSGE00): Get UIDs, GIDs, and Supplemental Groups" on page 28	z/OS UNIX file system
"getUMAP (IRRSUM00): Get UID-to-User-ID Mapping" on page 30	MVS BCP
"initACEE (IRRSIA00): Initialize ACEE" on page 32	z/OS kernel on behalf of servers that use pthread_security_np servers or __login, or MVS servers that do not use z/OS UNIX services
"initUSP (IRRSIU00): Initialize USP" on page 46	MVS BCP or z/OS UNIX servers
"makeFSP (IRRSFM00): Make IFSP" on page 49	z/OS UNIX file system or z/OS UNIX servers
"makeISP (IRRSMI00): Make IISP" on page 52	MVS BCP or z/OS UNIX task level processes
"make_root_FSP (IRRSRM00): Make Root IFSP" on page 55	DFSMS/MVS or z/OS UNIX servers
"query_file_security_options (IRRSQF00): Query File Security Options" on page 57	z/OS UNIX file system
"query_system_security_options (IRRSQS00): Query System Security Options" on page 59	MVS BCP
"R_admin (IRRSEQ00): RACF Administration API" on page 61	Tivoli
"R_audit (IRRSAU00): Provide an audit interface" on page 99	z/OS UNIX file system, MVS BCP, or z/OS UNIX servers
"R_cacheserv (IRRSCH00): Cache Services" on page 101	Policy Director
"R_chaudit (IRRSKA00): Change Audit Options" on page 108	z/OS UNIX file system or z/OS UNIX servers
"R_chmod (IRRSKF00): Change File Mode" on page 110	z/OS UNIX file system or z/OS UNIX servers

Table 1. Intended Use of RACF Callable Services (continued)

Callable Service	For Use By
"R_chown (IRRSCO00): Change Owner and Group" on page 112	z/OS UNIX file system or z/OS UNIX servers
"R_datalib (IRRSDL00): OCSF Data Library" on page 115	MVS BCP or z/OS UNIX servers
"R_dceauth (IRRSDA00): Check a User's Authority" on page 123	MVS BCP
"R_dceinfo (IRRSDI00): Retrieve or Set User Fields" on page 126	MVS BCP
"R_dcekey (IRRSDK00): Retrieve or Set a non-RACF Password" on page 131	MVS BCP
"R_dceruid (IRRSUD00): Determine the ID of a Client" on page 134	z/OS UNIX servers or MVS BCP
"R_exec (IRRSEX00): Set Effective and Saved UIDs/GIDs" on page 137	MVS BCP or z/OS UNIX task level processes
"R_fork (IRRSFK00): Fork a Process" on page 139	MVS BCP or z/OS UNIX task level processes
"R_getgroups (IRRS GG00): Get/Set Supplemental Groups" on page 142	MVS BCP or z/OS UNIX servers
"R_getgroupsbyname (IRRSUG00): Get Groups by Name" on page 144	MVS BCP
"R_IPC_ctl (IRRS CI00): Perform IPC Control" on page 147	MVS BCP or z/OS UNIX task level processes
"R_proxy serv (IRRS PY00): LDAP interface" on page 187	Policy Director
"R_ptrace (IRRS PT00): Ptrace Authority Check" on page 194	MVS BCP or z/OS UNIX task level processes
"R_setegid (IRRS EG00): Set Effective GID, Set All GIDs" on page 195	MVS BCP
"R_seteuid (IRRS EU00): Set Effective UID, Set All UIDs" on page 197	MVS BCP
"R_setfacl (IRRS CL00): Unix Access Control Lists" on page 199	z/OS UNIX file system or z/OS UNIX servers
"R_setgid (IRRS SG00): Set Group Name" on page 202	MVS BCP
"R_setuid (IRRS SU00): Set z/OS UNIX user identifier (UID)" on page 204	MVS BCP
"R_umask (IRRS MM00): Set File Mode Creation Mask" on page 210	MVS BCP or z/OS UNIX servers
"R_usermap (IRRS IM00): Map application user" on page 212	z/OS application servers

Note: In a server environment, work can be processed for more than one user in an address space. Callable services marked for use by z/OS UNIX servers provide task-level support for server applications. Callable services marked as having support for task level processes use task-level support when z/OS UNIX has indicated in the task's ACEE that this is a task level process. All other callable services assume that there is only one user per address space and provide only address-space-level support.

ck_access (IRRSKA00): Check Access

Function

The `ck_access` service determines whether the current process has the requested access to the element (directory or file) of a pathname whose IFSP, and ACL if it exists, is passed. It is called separately for each element.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user or any task if system user type is specified
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF authorization

1. If the audit function code in the CRED is access, the real z/OS UNIX user identifier (UID) and z/OS UNIX group identifier (GID) of the calling process are used on the authority checks. Otherwise, the effective UID and GID for the calling process are used.
2. If the calling user has auditor authority and the access requested is search, or the access requested is read for a directory, access is allowed. This lets an auditor set the auditor audit options on any file without requiring that the auditor be given search access rights to all directories.
3. If the CRED user type is system, IRRSKA00 allows any access except when the requested access is execute and no execute permission bits are set for the file. No UIDs are used in this case, because no process exists.
4. If the caller is not a superuser, the permission bits did not allow the requested access, *and* the audit function code is listed in Table 2 on page 10, an authorization check is performed on the corresponding resource in the UNIXPRIV class. If the authorization check is successful, the caller is treated as a superuser.

If Table 2 does not result in a UNIXPRIV authorization check, the caller is not a superuser, the permission bits did not allow the requested access, the file is a directory, and requested access is search, a read authorization check is performed on the SUPERUSER.FILESYS resource in the UNIXPRIV class. If the authorization check is successful, the caller is treated as a superuser.

If a matching ACL entry was encountered for the user, or for at least one of its groups, and the requested access was not granted, then substitute SUPERUSER.FILESYS.ACLOVERRIDE as the resource name in Table 2. If SUPERUSER.FILESYS.ACLOVERRIDE does not exist, the check is redriven for the SUPERUSER.FILESYS resource.

ck_access

Table 2. UNIXPRIV class resource names used in ck_access

Audit function code	Resource name	Access required
OPEN (for read or search), OPENDIR, READLINK, STAT, REALPATH, LSTAT, EACCESS(for read), ACCESS(for read; if real, effective and saved match)	SUPERUSER.FILESYS	READ
OPEN (for write), EACCESS(for write), ACCESS(for write; if real, effective and saved match)	SUPERUSER.FILESYS	UPDATE
LINK, MKDIR, RENAME, RMDIR, SYMLINK, UNLINK	SUPERUSER.FILESYS	CONTROL

5. If the user being checked is a superuser, IRRSKA00 allows any access except when the requested access is execute and no execute permission bits are set for the file. The user is considered a superuser if the selected UID is 0 or if the ACEE indicates trusted or privileged authority.
6. If the user is not system and is not a superuser, the permission bits and ACL (if one exists, and if the FSSEC class is active) for the file are checked to see if the access requested is allowed. If the selected UID matches the owner UID of the file, the owner permission bits are checked. If the UIDs don't match, the user ACL entries are checked. If the selected UID matches an ACL entry, the ACL entry bits are checked. If a matching ACL entry was not found for the user, the group bits and the group ACL entries are checked. The selected GID, and supplemental GIDs, are checked against the file owner GID and the group ACL entries, until a match is found which grants the requested access, or until all the GIDs have been checked. If no match was found, the other permission bits are checked, unless the user has the RESTRICTED attribute, the UNIXPRIV class is active, the resource named RESTRICTED.FILESYS.ACCESS is protected, and the user does not have at least READ access.
7. If the real, effective, and saved UID are the same, and if the real, effective, and saved GID are the same, UNIXPRIV will be checked for AFC_ACCESS.
8. For a detailed list of authorization steps for z/OS UNIX files and directories, see Appendix F, in the *z/OS Security Server RACF Security Administrator's Guide*.

Format

```
CALL IRRSKA00 (Work_area,  
              ALET, SAF_return_code,  
              ALET, RACF_return_code,  
              ALET, RACF_reason_code,  
              ALET, Requested_access_code,  
              ALET, FSP,  
              ALET, File_identifier,  
              ALET, CRED,  
              ALET, Name_flag  
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Requested_access_code

The name of a 1-byte field containing the requested access. The defined codes are:

- X'00' no access
- X'01' Execute access
- X'02' Write access
- X'03' Write and execute access
- X'04' Read access
- X'05' Read and execute access
- X'06' Read and write access
- X'07' Read, write, and execute access
- X'81' Search access (against a directory)
- X'87' Any access

FSP

The name of the IFSP for the file being accessed.

File_Identifier

The name of a 16-byte area containing a unique identifier of the file.

CRED

The name of the CRED structure for the current file system syscall. See the *z/OS Security Server RACF Data Areas*. The CRED contains a pointer to the ACL, if one exists.

Name_flag

The name of a byte indicating which pathname and file name is being checked. The byte contains one of these values:

- 0 Use the CRED_name_flag to determine pathname being checked.
- 1 The old (or only) name is being checked.
- 2 The new name is being checked.

Return and reason codes

IRRSKA00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.

ck_access

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
4	0	0	RACF is not installed.
8	8	4	The user is not authorized to access the file.
8	8	32	The CRED user type is not supported.

Usage notes

1. This service is intended only for use by a z/OS UNIX file system and by z/OS UNIX servers. The service contains support for z/OS UNIX servers, but can not be directly invoked by a z/OS UNIX server.
2. The access checks performed are POSIX file permission checks defined in POSIX 1003.1.
3. If the audit function code in the CRED is access or eaccess, no audit record is written. Access checking only tests whether a process would have access if it were running with its real UID. Eaccess checking only tests whether a process would have access with its effective UID. Neither gives access to the file.
4. If the calling syscall is not access (for real or effective UID), an audit record is optionally written, depending on the audit options in effect for the system.
5. The caller must pass in the address of the object's access ACL in the CredAclPtr field.

Related services

R_chmod, R_chown, R_setfacl

ck_file_owner (IRRSKF00): Check File Owner

Function

The **ck_file_owner** service checks whether the calling process is a superuser or is the owner of the file represented by the input

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. The ck_file_owner service checks whether the calling process is a superuser.
2. The ck_file_owner service checks whether the calling process is the owner of the file represented by the input IFSP.
3. A process is the owner of a file if the process's effective UID is equal to the file's owner UID.

Format

```
CALL IRRSKF00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, FSP,
               ALET, File_identifier,
               ALET, CRED
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a full word in which the SAF router returns the SAF return code.

RACF_return_code

The name of a full word in which the service routine stores the return code.

RACF_reason_code

The name of a full word in which the service routine stores the reason code.

FSP

The name of the IFSP for the file to be checked.

File_Identifier

The name of a 16-byte area containing a unique identifier of the file.

CRED

The name of the CRED structure for the current file system syscall. See *z/OS Security Server RACF Data Areas*.

Return and Reason Codes

IRRSKF00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.

ck_file_owner

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
8	8	4	The user is not a superuser or the file owner.
8	8	12	An internal error occurred during RACF processing.
8	8	32	The CRED user type is not supported.

Usage Notes

1. This service is intended only for use by a z/OS UNIX file system and by z/OS UNIX servers. The service contains support for z/OS UNIX servers, but can not be directly invoked by a z/OS UNIX server.
2. An audit record is optionally written, depending on the audit options in effect for the system.

Related Services

None

ck_IPC_access (IRRSKI00): Check IPC Access

Function

The `ck_IPC_access` service determines whether the current process has the requested access to the interprocess communication (IPC) key or identifier whose IPC security packet (IISP) is passed.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user/any task if system user type is specified
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	None
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. The access checks performed are XPG4 IPC permission checks defined in XPG4 System Interfaces and Headers, as follows:
 - The effective z/OS UNIX user identifier (UID) and z/OS UNIX group identifier (GID) for the calling process is used for all access checks.
 - If the CREI user type is system, IRRSKI00 allows any access. No UIDs or GIDs are used in this case because no process exists.

- If the user being checked is a superuser, IRRSKI00 allows any access. The user is considered a superuser if the selected UID is 0 or if the ACEE indicates trusted or privileged authority.
- If the user is not system and is not a superuser, the permission bits for the IPC key are checked to see if the access requested is allowed. If the effective UID matches either the owner UID or creator's UID of the IPC key, the USER permission bits are checked. If the UIDs do not match, the owner GID and creator's GID of the IPC key are checked against the user's effective GID and the user's supplemental group list GIDs. If any one matches, the GROUP permission bits are checked. If the UIDs and GIDs don't match, the OTHER permission bits are checked.

Format

```
CALL IRRSKI00 (Work_area,
              ALET, SAF_return_code,
              ALET, RACF_return_code,
              ALET, RACF_reason_code,
              ALET, Requested_access_code,
              ALET, ISP,
              ALET, CREDIPC
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Requested_access_code

The name of a one-byte field containing the requested access. The defined codes are:

```
X'00'   No access
X'02'   Write access (or alter access)
X'04'   Read access
X'06'   Read and write access
```

ISP

The name of the IISP for the key being accessed.

CREDIPC

The name of the CREI structure for the current IPC system callable service. Use the CREI to determine the IPC identifier and IPC key being used. See *z/OS Security Server RACF Data Areas*.

Return and Reason Codes

IRRSKI00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The user is not authorized to access the IPC mechanism.
8	8	32	CREI user type is not supported.

Usage Notes

1. This service is intended for use only by the MVS BCP.
2. An audit record is optionally written, depending on the audit options in effect for the system.
3. This service uses task level support when z/OS UNIX has indicated in the task's ACEE that this is a task level process.

Related Services

makeISP, R_IPC_ctl

ck_owner_two_files (IRRSC200): Check Owner of Two Files

Function

The `ck_owner_two_files` service checks whether the calling process is a superuser or is the owner of either of the file/directory, or directory/directory entry pair represented by input values FSP1 and FSP2.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. A process is the owner of the file if the process's effective OS/390 UNIX user identifier (UID) is equal to the file's owner UID.

- If the caller is not superuser nor the owner, and the audit function code is listed in Table 3, an authorization check is performed on the corresponding resource name in the UNIXPRIV class. If the authorization check is successful, the caller is treated as a superuser.

Table 3. UNIXPRIV class resource names used in ck_owner_two_files

Audit function code	Resource name	Access required
RENAME, RMDIR, UNLINK	SUPERUSER.FILESYS	CONTROL

Format

```
CALL IRRSC200 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, FSP1,
               ALET, FSP2,
               ALET, File_identifier_1,
               ALET, File_identifier_2,
               ALET, CRED
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

FSP1

The name of the IFSP for the first file, directory, or directory entry to be checked. If FSP1 is a file, FSP2 must be a directory. If FSP1 is a directory entry, FSP2 must be a directory.

FSP2

The name of the IFSP for the second file, directory, or directory to be checked. If FSP2 is a file, FSP1 must be a directory. If FSP2 is a directory entry, FSP1 must be a directory.

File_identifier_1

The name of a 16-byte area containing a unique identifier of the first file to be checked.

File_identifier_2

The name of a 16-byte area containing a unique identifier of the second file to be checked.

ck_owner_two_files

CRED

The name of the CRED structure for the current file system syscall. See *z/OS Security Server RACF Data Areas*.

Return and Reason Codes

IRRSC200 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The caller is not a superuser or the file owner.
8	8	12	An internal error occurred during RACF processing.
8	8	32	CRED user type is not supported.

Usage Notes

1. This service is intended only for use by a z/OS UNIX file system and by z/OS UNIX servers. The service contains support for z/OS UNIX servers, but cannot be directly invoked by a z/OS UNIX server.
2. An audit record is optionally written, depending on the audit options in effect for the system.

Related Services

None

ck_priv (IRRSKP00): Check Privilege

Function

The `ck_priv` service checks whether the calling process is a superuser.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. A superuser is a user whose process has an effective UID of 0 or has RACF trusted or privileged authority.
2. If the caller is not superuser and the audit function code is listed in Table 4, an authorization check is performed on the corresponding resource name in the UNIXPRIV class. If the authorization check is successful, the caller is treated as a superuser.

Table 4. UNIXPRIV class resource names used in ck_priv

Audit function code	Resource name	Access required
MOUNT(nosetuid), UNMOUNT(nosetuid), CHMOUNT(nosetuid),	SUPERUSER.FILESYS.MOUNT	READ
MOUNTSETUID, UNMOUNTSETU, CHMOUNT(setuid)	SUPERUSER.FILESYS.MOUNT	UPDATE
QUIESCE(nosetuid), UNQUIESCE(nosetuid)	SUPERUSER.FILESYS.QUIESCE	READ
QUIESCESETU, UNQUIESCESU	SUPERUSER.FILESYS.QUIESCE	UPDATE
PFSCCTL	SUPERUSER.FILESYS.PFSCCTL	READ
SETPRIORITY, NICE	SUPERUSER.SETPRIORITY	READ
VREGISTER	SUPERUSER.FILESYS.VREGISTER	READ

Format

```
CALL IRRSKP00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Audit_function_code
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

ck_priv

Audit_function_code

The name of a fullword containing a function code identifying the system call function being processed. See *z/OS Security Server RACF Data Areas* for a list of the defined codes.

Return and Reason Codes

IRRSKP00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The user is privileged.
4	0	0	RACF is not installed.
8	8	4	The user is not privileged.
8	8	12	An internal error occurred during RACF processing.

Usage Notes

1. This service is intended for use only by the MVS BCP, a z/OS UNIX file system, and by z/OS UNIX servers. The service contains support for z/OS UNIX servers, but can not be directly invoked by a z/OS UNIX server.
2. An audit record is written.

Related Services

None

ck_process_owner (IRRSK00): Check Process Owner

Function

The `ck_process_owner` service checks whether the calling process is the owner of the target process.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. For request types 2, 3, and 4, IRRSKO00 checks whether the caller has superuser authority or is the owner of the target process, and returns a return and reason code indicating the result.
2. The caller is an owner of a process if either the real or effective z/OS UNIX user identifier (UID) of the calling process is equal to either the real or saved UID passed in the *Target_process_UIDs* parameter area.
3. If the caller is not superuser nor the process owner, and the request type is listed in Table 5, an authorization check is performed on the corresponding resource name in the UNIXPRIV class. If the authorization check is successful, the caller is treated as a superuser.

Table 5. UNIXPRIV class resource names used in ck_process_owner

Request type	Resource name	Access required
2	SUPERUSER.PROCESS.KILL	READ
3	SUPERUSER.PROCESS.GETPSENT	READ

Format

```
CALL IRRSKO00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Request_type,
               ALET, Target_process_UIDs,
               ALET, Target_PID,
               ALET, Signal_code
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Request_type

The address of a byte containing a request type. The defined types are:

- 1 - audit-only request from kill. It is used when a SIGCONT signal is being sent to a process in the same session as the signalling process.
- 2 - kill request
- 3 - getpsent request
- 4 - open_tty request

ck_process_owner

Target_process_UIDs

The address of a 3-word area containing the real, effective, and saved z/OS UNIX user identifiers (UIDs) (in that order) for the target process.

Target_PID

The name of a fullword containing the PID of the target process.

Signal_code

The address of a word containing a code that identifies the type of signal being sent. This code is used only for auditing. The signal code values are defined in the z/OS UNIX macro BPXYSIGH. This parameter is ignored for request type 3.

Return and Reason Codes

IRRSKO00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The caller is not the owner of the target process.
8	8	8	The request type is not valid.
8	8	12	An internal error occurred during RACF processing.

Usage Notes

1. This service is intended only for use by the MVS BCP.
2. An audit record is optionally written, depending on the audit options in effect for the system.
3. This service uses task level support when z/OS UNIX has indicated in the task's ACEE that this is a task level process.

Related Services

None

clear_setid (IRRSCS00): Clear Set ID

Function

The `clear_setid` service clears the S_ISUID, S_ISGID, and S_ISVTX bits for the file passed as input.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR

Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

None

Format

```
CALL IRRSCS00 (Work_area,
               ALET,SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, FSP,
               ALET, File_identifier,
               ALET, CRED
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

FSP

The name of the IFSP in which the S_ISUID, S_ISGID, and S_ISVTX bits are to be cleared.

File_Identifier

The name of a 16-byte area containing a unique identifier of the file.

CRED

The name of the CRED structure for the current file system syscall. See *z/OS Security Server RACF Data Areas*.

clear_setid

Return and Reason Codes

IRRSCS00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	32	The CRED user type is not supported.

Usage Notes

1. This service is intended only for use by an z/OS UNIX System Services file system and by z/OS UNIX System Services servers. The service contains support for z/OS UNIX System Services servers, but can not be directly invoked by an z/OS UNIX System Services server.
2. The caller is responsible for preserving the updated IFSP.
3. If either bit was on, an audit record is optionally written.

Related services

R_chmod, R_exec

deleteUSP (IRRSDU00): Delete USP

Function

The **deleteUSP** service deletes the security environment for the calling process. The caller can continue as an MVS user, but is no longer an z/OS UNIX process.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

None

Format

```
CALL IRRSDU00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Return and Reason Codes

IRRSDU00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.

Usage Notes

1. This service is intended only for use by the MVS BCP and by z/OS UNIX System Services servers. This service can be directly invoked by an z/OS UNIX System Services server.
2. An audit record is optionally written, depending on the audit options in effect for the system.

Related Services

initUSP

getGMAP (IRRSGM00): Get GID-to-Group-Name Mapping

Function

The **getGMAP** service returns the z/OS UNIX group identifier (GID) or group name corresponding to the input group name or GID, based on the setting of an input lookup flag.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

None

Format

```
CALL IRRSGM00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Flag,
               ALET, GID,
               ALET, group_name
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Flag

The name of a word containing the lookup option:

X'00000000'

search by z/OS UNIX group identifier (GID), return group name

X'00000001'

search by group name, return GID

GID

The name of a fullword for a z/OS UNIX group identifier (GID). The GID is either input or output in this word, depending on the flag parameter.

Group_name

The name of an 8-byte area for the group name. The group name is left-justified and padded with blanks and is either input or output in the area, depending on the flag parameter.

Return and Reason Codes

IRRSGM00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	If search by GID: GID is not defined. If search by group name: The current group's profile has no OMVS segment.
8	8	8	The group name is not defined.
8	8	12	An internal error occurred during RACF processing.
8	8	16	Recovery could not be established.
8	8	20	OMVS segment of the current group's profile has no GID.
8	8	24	The maximum number of file descriptors (OPEN MAX) are currently open in the calling process. Note: RACF does not issue this return code, but other security products may.
8	8	28	The maximum allowable number of files is currently open in the system. Note: RACF does not issue this return code, but other security products may.

Usage Notes

- This service is intended only for use by the MVS BCP.
- If getGMAP is given a group name as input and the corresponding GROUP profile has no OMVS segment, getGMAP checks the BPX.DEFAULT.USER profile in the FACILITY class. This profile may contain a group name in its application data field that provides a default OMVS segment. If this default is found, its z/OS UNIX group identifier (GID) is returned to the issuer of getGMAP.
- Check if any logrec entry has been created to ensure getGMAP service was being run successfully. Refer to *z/OS Security Server RACF Diagnosis Guide* for detailed logrec information.

Related Services

None

get_uid_gid_supgrps (IRRSGE00): Get UIDs, GIDs, and Supplemental Groups

Function

The **get_uid_gid_supgrps** service gets the real, effective, and saved z/OS UNIX user identifiers (UIDs) and z/OS UNIX group identifiers (GIDs), and the supplemental groups from the USP.

Because the size of the supplemental group list varies, IRRSGE00 checks the input group count before putting supplemental GIDs in the grouplist area. See **Group_count** under “Parameters” on page 29 for more information.

The GIDs are not explicitly added to or deleted from the supplemental group list. A GID is in this list if the user was a member of the group when the user’s ACEE was created through a RACROUTE REQUEST=VERIFY request and if the GID was assigned to the group before the **initUSP** service was performed for the process.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

None

Format

```
CALL IRRSGE00 (Work_area,  
              ALET, SAF_return_code,  
              ALET, RACF_return_code,  
              ALET, RACF_reason_code,  
              ALET, RACF_work_area,  
              ALET, User_key,  
              ALET, Group_count,  
              ALET, Group_list,  
              ALET, Number_of_GIDs,  
              ALET, Output_UIDs,  
              ALET, Output_GIDs  
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

RACF_work_area

The name of a 1024-byte work area for RACF use.

User_key

The name of a byte containing the user's key. This key is used to store into the output grouplist area. The key is in the four high-order bits of the byte.

Group_count

The name of a word containing the number of z/OS UNIX group identifier (GID) entries that can be stored in the *Grouplist* area. If *Group_count* is:

1. 0, the *Grouplist* area is not used. IRRSGE00 returns the total supplemental GID count of the current process in the *Number_of_GIDs* parameter.
2. Less than the total supplemental GID count:
 - a. An error code is returned.
 - b. The GIDs of the supplemental groups for the current process are put into the *Grouplist* area, which can only accommodate the number of GIDs specified in the *Group_count* parameter.
 - c. The count of the supplemental GIDs actually placed in the *Grouplist* area is returned in the *Number_of_GIDs* parameter.
 - d. The *Group_count* field is set to the total supplemental GID count of the current process.
The supplemental groups in the *Grouplist* area are listed in the same order as the group connections shown in the output of the LISTUSER command.
3. Greater than or equal to the total supplemental z/OS UNIX group identifier (GID) count:
 - a. The GIDs of the supplemental groups for the current process are put into the *Grouplist* area.
 - b. The supplemental GID count of the current process is put into the *Number_of_GIDs* parameter.

Grouplist

The name of an area in which the GIDs of the supplemental groups for a process are returned. The *Group_count* parameter indicates the number of entries this area can contain. The GIDs are returned as consecutive 4-byte entries.

get_uid_gid_supgrps

Number_of_GIDs

The name of a word in which the number of GIDs put in the *Grouplist* area is returned.

Output_UIDs

The name of a 3-word area in which, respectively, the real, effective, and saved z/OS UNIX user identifiers (UIDs) are returned.

Output_GIDs

The name of a 3-word area in which, respectively, the real, effective, and saved z/OS UNIX group identifiers (GIDs) are returned.

Return and Reason Codes

IRRSGE00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	<i>Group_count</i> is less than the number of supplemental groups (see item 2 on page 29 under the Group_count parameter).
8	8	8	The grouplist address is not valid.
8	8	12	An internal error occurred during RACF processing.

Usage Notes

- This service is intended only for use by z/OS UNIX. The service which executes in the primary address space contains support that accesses the home address space task control block and address space control block for the requested data.
- In order to support multiple processes in one address space, this function needs to return the requested data from either the task control area or the address space control area. The task control area is accessed before the address space control area.

Related Services

None.

getUMAP (IRRSUM00): Get UID-to-User-ID Mapping

Function

The **getUMAP** service returns the z/OS UNIX user identifier (UID) or user ID corresponding to the input user ID or UID, based on the setting of an input lookup flag.

Requirements

Authorization:

Any PSW key in supervisor state

Dispatchable unit mode:

Task of z/OS UNIX user

Cross memory mode:

PASN = HASN

AMODE:

31

RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

None

Format

```
CALL IRRSUM00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Flag,
               ALET, UID,
               ALET, Userid
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Flag

The name of a word containing the lookup option:

X'00000000'

search by z/OS UNIX user identifier (UID), return user ID

X'00000001'

search by user ID, return z/OS UNIX user identifier (UID)

UID

The name of a fullword for a z/OS UNIX user identifier (UID). The UID is either input or output in this word, depending on the flag parameter.

getUMAP

Userid

The name of an 8-byte area for the user ID. The user ID is left-justified and padded with blanks, and is either input or output in the area depending on the flag parameter.

Return and Reason Codes

IRRSUM00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	If search by UID: UID is not defined. If search by user ID: The user's profile has no OMVS segment.
8	8	8	User ID is not defined.
8	8	12	An internal error occurred during RACF processing
8	8	16	Recovery could not be established.
8	8	20	The OMVS segment of the user's profile has no UID.
8	8	24	The maximum number of file descriptors (OPEN MAX) are currently open in the calling process. Note: RACF does not issue this return code, but other security products may.
8	8	28	The maximum allowable number of files is currently open in the system. Note: RACF does not issue this return code, but other security products may.

Usage Notes

- This service is intended only for use by the MVS BCP.
- If getUMAP is given a user ID as input, and the corresponding USER profile has no OMVS segment, getUMAP checks the BPX.DEFAULT.USER profile in the BPX.DEFAULT.USER profile in the FACILITY class. This profile may contain a user ID in its application data field that provides a default OMVS segment. If this default is found, its UID is returned to the issuer of getUMAP.
- Check if any logrec entry has been created to ensure getUMAP service was being run successfully. Refer to *z/OS Security Server RACF Diagnosis Guide* for detailed logrec information.

Related Services

None.

initACEE (IRRSIA00): Initialize ACEE

Function

The **initACEE** service provides an interface for creating and managing RACF security contexts through the z/OS UNIX System Services `pthread_security_np`

service, __login service, or by other MVS server address spaces that do not use z/OS UNIX services. This service also provides an interface for registering and deregistering certificates through the z/OS UNIX System Services __security service. It also provides an interface for querying a certificate to determine if it is associated with a user ID.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. The words containing the ALETs must be in the primary address space.

Linkage Conventions

The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, the last word in the parameter list must have a 1 in the high-order (sign) bit.

RACF Authorization

1. If the function_code indicates that a certificate is to be registered or deregistered, initACEE will perform the following authority checks:
 - To register a certificate with the current user ID, the caller must be RACF SPECIAL or have at least READ authority to the IRR.DIGTCERT.ADD resource in the FACILITY class.
 - To deregister a certificate with the current user ID, the caller must be RACF SPECIAL or have at least READ authority to the IRR.DIGTCERT.DELETE resource in the FACILITY class.
 - To register a certificate as a CERTAUTH certificate, the caller must be RACF SPECIAL or have at least CONTROL authority to the IRR.DIGTCERT.ADD resource in the FACILITY class.
2. If the function_code indicates that an ACEE is to be created or a certificate is to be queried and the service determines that the user ID to use is specified in the hostIdMappings extension of the input certificate, the caller's authority to the IRR.HOST.(host-name) resource in the SERVAUTH class is checked. (The value for host-name is specified in the hostIdMappings extension.) The resource must exist and the caller must have READ authority to it, otherwise the extension is ignored.

Note: To determine the caller, the current TCB is checked for an ACEE. If one is found, the authority of that user is checked. If there is no ACEE associated with the current TCB, the ACEE associated with the address space is used to locate the user ID.

Format

```
CALL IRRSIA00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               Function_code,
               Attributes,
               RACF_userid,
               ACEE_ptr,
               APPL_id,
               Password,
               Logstring,
               Certificate,
               ENVR_in,
               ENVR_out,
               Output_area,
               X500name,
               Variable_list
            )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Function_code

The name of a 1 byte area containing the function code

- X'01' Create an ACEE.
- X'02' Delete an ACEE.
- X'03' Purge all managed ACEEs.
- X'04' Register a certificate
- X'05' Deregister a certificate
- X'06' Query a certificate

Attributes

The name of a 4 byte area containing information about the function to be performed. Zero or more attributes can be set. (See Table 6 on page 37 for the values allowed for the Attributes parameter.)

RACF_userid

The name of a 9 byte area that consists of a 1 byte length field followed by up to 8 characters. It must be specified in upper case. If not specified, the length must equal 0.

ACEE_ptr

The name of a 4 byte area that contains the ACEE address.

APPL_id

The name of a 9 byte area that consists of a 1 byte length field followed by the name of the application to be used if verifying the user's authority to access the application. This saves the application from having to do a separate authorization check. When using certificate mapping profiles, the application name is also used as part of the additional criteria in determining a user ID when a certificate is passed to initACEE. It must be specified in upper case. If not specified, the length must equal zero.

Password

The name of a 9 byte area that consists of a 1 byte length field followed by the password or PassTicket provided by the user. It must be specified in upper case. If not specified, the length must equal zero.

Logstring

The name of an area that consists of a 1 byte length field followed by character data to be written to the system-management-facilities (SMF) data set, together with any RACF audit information, if logged. If not specified, the length must equal zero.

Certificate

The name of an area that consists of a 4 byte length field followed by a digital certificate. If not specified, the length must equal 0; or the end of the parameter list must be indicated by the setting of the high order bit in the address of the previous parameter. The certificate must be a single DER encoded X.509 certificate. For the registration and deregistration functions, PKCS #7, PEM, or Base64 encoded certificates are also allowed.

ENVR_in

The name of the data structure that contains the information necessary to re-create a security environment. The data structure must have the format shown in Table 7 on page 38. See the ENVR_out parameter for additional information on this data structure and the ENVR object to which it points. The structure must reside on a doubleword boundary.

While the format of the data structure pointed to by ENVR_in is known to the initACEE invokers, the content of the object itself is determined by the external security product.

The input for this parameter can be the output from a previous initACEE with the ENVR_out parameter specified, or from RACROUTE REQUEST=VERIFY or REQUEST=EXTRACT, with the ENVROUT parameter specified.

If ENVR_in is not specified, the ENVR object length must equal 0, or the end of the parameter list must be indicated by the setting of the high order bit in the address of a previous parameter. ENVR_in should not be specified when requesting that an ENVR object be returned (INTA_ENVR_RET).

For more information about the ENVR data structure, see *z/OS Security Server RACROUTE Macro Reference*.

ENVR_out

The name of the data structure to contain the security environment that was just created. The data structure must have the format shown in Table 7 on page 38. This data structure describes the storage location for the ENVR object that is created as part of this initACEE create request.

initACEE

While the format of the data structure pointed to by ENVR_out itself is known to the initACEE invokers, the content of the object itself is determined by the external security product.

The ENVR object storage area can be supplied by the caller or obtained by RACF. If supplied by the caller, it must be on a doubleword boundary and be associated with the job step task. If RACF obtains the storage area, it is on a doubleword boundary and is associated with the job step task. The storage is allocated based on the mode of the caller (LOC=ANY for 31-bit callers and LOC=24 for 24-bit callers).

Storage for the ENVR object is obtained and freed in the subpool and key specified by the caller in the ENVR_out data structure. For additional details on specifying the ENVR object storage area length and address, see Table 8 on page 38.

Since the ENVR object length is returned to the caller, the ENVR object can be moved from one storage area to another. It is intended for use on subsequent initACEEs with the ENVR_in parameter, or on RACROUTE REQUEST=VERIFY with the ENVRIN parameter, as input when rebuilding a user's security environment. It should not be saved for a long period or passed to another system that does not share the same RACF database.

If the Attributes parameter indicates that an ENVR object should be returned (INTA_ENVR_RET), then this parameter must be specified with at a minimum the values for the subpool and key fields.

For more information about the ENVR data structure, see *z/OS Security Server RACROUTE Macro Reference*.

Output_area

The name of a fullword in which the service routine stores the address of an area containing data about the user. The output area is obtained in the primary address space, in subpool 229, and must be freed by the caller of initACEE. The following data is returned; the area returned is mapped by macro IRRPOUSP (Ousp):

- TSO user ID
- z/OS UNIX user identifier (UID) of user
- z/OS UNIX group identifier (GID) of current group
- Home directory path name
- Initial program path name
- User limits (when Ousp version is greater than 0)

If the Attributes parameter indicates that an Ousp should be returned (INTA_USP and INTA_Ousp_RET), then this parameter must be specified. If the Attributes do not indicate that an Ousp should be returned, then the fullword must equal 0, or the end of the parameter list must be indicated by the setting of the high order bit in the address of a previous parameter.

X500name

The name of a fullword in which the service routine stores the address of the X500 name pair data structure if the function code indicates a certificate is being queried, and the attributes indicate that an X500 name pair should be returned. The X500 name pair data structure is obtained in the primary address space, in subpool 229, and must be freed by the caller of initACEE.

If the function code indicates that an ACEE is to be created, and the RACF_userid parameter is specified, X500name can supply the name of a

fullword containing the address of the X500 name pair to be associated with the ACEE. The X500 name pair should previously have been obtained, along with the RACF user ID, by querying a certificate using initACEE. Both the issuer's name and subject's name must be supplied, and the length of each must be in the range 1 to 255 to prevent a parameter list error. If a valid X500 name pair is supplied, the ACEE created will point to a copy of the name pair, and it will subsequently be used in auditing.

Variable_list

The name of the data structure that contains the additional criteria to be used to determine the user ID associated with the certificate supplied to initACEE. The criteria value data structure is a 4-byte number of value entries, followed by that number of entries. Each value entry consists of an 8-byte value name, a 4-byte value length, and the value. The value name must be padded on the right with blanks if it is less than 8-bytes. The value length must be in the range of 1 to 255. If it is outside of this range, a parameter list error will result. A maximum of 10 values may be specified. If the number of values is greater than 10, a parameter list error will result.

Variable names should be meaningful to the caller of initACEE. Making the 3 character prefix associated with the product calling initACEE part of the variable name will ensure that it is unique. For example, assume RACF implemented a server that calls initACEE for its clients. It will pass a variable, IRRSLVL, which has 2 values. The values are LOW and HIGH. LOW is if the user is accessing the server from the internet and HIGH is if the user is accessing the server from the intranet. The variable_list containing the variable name and its value, LOW or HIGH, is passed to initACEE, along with the certificate supplied by the user. The value of the variable will be used as additional criteria in selecting which user ID the certificate maps to. All callers of initACEE should document their variable names, and the values they pass for each name, in their product documentation.

All value names and values should be upper case. Do not specify the APPLID or SYSID criteria values in the variable_list. These are determined from the APPL_id parameter and MVS control; blocks, respectively. If they are specified in the variable_list, that specification will be ignored.

This parameter is ignored unless the certificate parameter is specified, and the function code indicates that an ACEE is to be created, or that the certificate is to be queried to find a user ID. If the certificate is defined to RACF in the DIGTCERT class, additional criteria will not be used, and the variable_list values will be ignored. If the certificate is not defined in the DIGTCERT class, the values in the variable list will be used along with APPL_id and SYSID to look for an associated user ID using the DIGTNMAP and DIGTCRIT classes if these classes are active and have been RACLISTed with SETROPTS.

Table 6. Values Allowed for Attributes Parameter

INTA_MANAGED	X'80000000'	Create an ACEE for the user ID that is cached by RACF.
INTA_USP	X'40000000'	Create a USP for the user ID
INTA_TASK_LVL	X'20000000'	For create function code, create an ACEE and attach to the current TCB. For delete function code, delete the ACEE attached to the current TCB.
INTA_UNAUTH_CLNT	X'10000000'	Create an ACEE for an unauthenticated client.

Table 6. Values Allowed for Attributes Parameter (continued)

INTA_AUTH_CLNT	X'08000000'	Create an ACEE for an authenticated client.
INTA_MGS_SUPP	X'04000000'	Suppress RACF messages produced as a result of creating a user's security context.
INTA_ENVR_RET	X'02000000'	Return an ENVR object for the ACEE created by this request.
INTA_NO_TIMEOUT	X'01000000'	Create a managed ACEE that does not time out. When this bit and INTA_MANAGED are set for the creation of a new managed ACEE, the ACEE is cached and does not expire after 5 minutes.
INTA_OUSP_RET	X'00800000'	Return an OUSP in the output area
INTA_X500_RET	X'00400000'	Return an X500 name pair

Table 7. ENVR Data Structure

Description	Length (Bytes)	ENVR_out Usage	ENVR_in Usage
ENVR object length	4	Output	Input
ENVR object storage area length	4	Input/Output (see Table 8)	Input
ENVR object storage area address	4	Input/Output (see Table 8)	Input
ENVR object storage area subpool	1	Input	n/a
ENVR object storage area key	1	Input	n/a

Table 8. ENVR_out Storage Area Processing

ENVR Object Storage Area Length	ENVR Object Storage Area Address	Result
Zero	Any value	RACF obtains storage size needed to contain ENVR object and sets ENVR object storage area length and address fields.
Nonzero	Zero	RACF obtains storage size specified or minimum needed to contain ENVR object and sets ENVR object storage area length and address fields.
Nonzero	Nonzero	RACF uses the area provided if large enough to contain ENVR object. If too small, RACF freemains the area, obtains a larger area, and sets ENVR object storage area length and address fields.

Table 9. X500 Name Pair Data Structure

Offset	Length(Bytes)	Description
0	4	Length of name pair data structure
4	2	Length of issuer's name (1 to 255)
6	2	Length of subject's name (1 to 255)
8	1 to 255	Issuer's distinguished name
*	1 to 255	Subject's distinguished name

Table 10. Criteria Value Data Structure

Offset	Length(Bytes)	Description
0	4	Number of value entries
4	8	Value name
12(C)	4	Value length (1 to 255)
16(10)	1 to 255	Value

Return and Reason Codes

IRRSIA00 returns the following values in the reason and return code parameters:

Table 11. initACEE Create Return Codes

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	Parameter list error occurred.
8	8	8	An internal error occurred during RACF processing.
8	8	12	Recovery environment could not be established.
8	8	16	User ID is not defined to RACF.
8	8	20	Password or Pass Ticket is not valid.
8	8	24	Password is expired.
8	8	28	User ID is revoked or user access to group is revoked.
8	8	32	User is not authorized.
8	8	36	Certificate is not valid.
8	8	40	Either no user ID or userid mapping is defined for this certificate or the status of the certificate or mapping is NOTRUST, or there are no mapping profiles associated with the certificate. Mapping profiles can either be defined as a Certificate Name Filtering profile or as SERVAUTH profiles that are used for HostID Mappings. See Usage Note number 37.

initACEE

Table 11. *initACEE Create Return Codes (continued)*

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
8	12	InitUSP reason code	initUSP failed. See initUSP reason codes in "Return and Reason Codes" on page 48.

Table 12. *initACEE Delete Return Codes*

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	Parameter list error occurred.
8	8	8	An internal error occurred during RACF processing.
8	8	12	Recovery environment could not be established.
8	8	16	An attempt was made to delete the server address space ACEE before invoking initACEE to purge all managed ACEEs.

Table 13. *initACEE Purge Return Codes*

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	Parameter list error occurred.
8	8	8	An internal error occurred during RACF processing.
8	8	12	Recovery environment could not be established.
8	8	16	There are managed ACEEs that are still in use.

Table 14. *initACEE Register and Deregister Return Codes*

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	Parameter list error occurred.
8	8	8	An internal error occurred during RACF processing
8	8	12	Recovery environment could not be established.
8	8	16	The user is not authorized.
8	8	20	The certificate does not meet RACF requirements.
8	8	24	The certificate is defined for another user.

Table 14. *initACEE Register and Deregister Return Codes (continued)*

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
8	8	28	RESERVED
8	8	32	RESERVED
8	8	36	The certificate is not valid.

Table 15. *initACEE Query Return Codes*

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	Parameter list error occurred.
8	8	8	An internal error occurred during RACF processing.
8	8	12	Recovery environment could not be established.
8	8	16	RESERVED
8	8	20	RESERVED
8	8	24	RESERVED
8	8	28	RESERVED
8	8	32	RESERVED
8	8	36	This certificate is not valid.
8	8	40	No user ID is defined for this certificate.

Table 16. *Parameter Usage*

Parameter	Create ACEE	Delete ACEE	Purge ACEE	Reg/Dereg Certificate	Query Certificate
SAF_return_code	Output	Output	Output	Output	Output
RACF_return_code	Output	Output	Output	Output	Output
RACF_reason_code	Output	Output	Output	Output	Output
Function_code	Input	Input	Input	Input	Input
Attributes	Input	Input	n/a	n/a	Input
RACF_userid	Input	n/a	n/a	n/a	Output
ACEE_ptr	Output	Input	n/a	n/a	n/a
APPL_id	Input	n/a	n/a	n/a	n/a
Password	Input	n/a	n/a	n/a	n/a
Logstring	Input	n/a	n/a	n/a	n/a
Certificate	Input	n/a	n/a	Input	Input

initACEE

Table 16. Parameter Usage (continued)

Parameter	Create ACEE	Delete ACEE	Purge ACEE	Reg/Dereg Certificate	Query Certificate
ENVR_in	Input	n/a	n/a	n/a	n/a
ENVR_out	Input/ Output See Table 7 on page 38	n/a	n/a	n/a	n/a
Output_area	Output	n/a	n/a	n/a	n/a
X500name	Input	n/a	n/a	n/a	Output
Variable_list	Input	n/a	n/a	n/a	Input

Usage Notes

1. This service is only intended for use by the z/OS UNIX kernel or by other MVS servers that do not use z/OS UNIX.
2. This service can only be used by supervisor state callers.
3. An ALET must be specified for the SAF_return_code, RACF_return_code, and RACF_reason_code parameters.
4. When ACEEs are created by initACEE, the following information is used on the RACROUTE REQUEST=VERIFY:
 - Password, if verifying a password
 - Appl_id, if verifying authority to an application
 - Logstring, if any audit records are created as a result of authenticating the user ID
 - LOC=ANY. If the caller is running in 31-bit address mode, the ACEE may be allocated above the 16MB line.
 - Subpool of the address space ACEE is used for the SUBPOOL keyword
 - ENVR_OUT, if an ENVR object data structure address was supplied by the ENVR_out parameter.
 - X500name, if the RACF_userid and X500_name parameters were specified, or if a certificate was provided as input and an associated user ID was found using the DIGTNMAP class profiles.
5. When creating an ACEE, statistics are updated on the first request per day for each user ID.
6. Audit records are written only in the following situations:
 - a. An ACEE is to be created and a password has been specified that is not the user's current password.
 - b. An ACEE is to be created and a PassTicket has been specified that does not evaluate.
 - c. An ACEE is to be created and the user ID has been revoked.
 - d. A certificate is to be registered, and the user is not authorized to the FACILITY class resource IRR.DIGTCERT.ADD.
 - e. A certificate is to be deregistered and the user is not authorized to the FACILITY class resource IRR.DIGTCERT.DELETE.
 - f. A certificate is successfully registered or deregistered, and SETROPTS AUDIT(USER) is in effect, or UAUDIT is in effect for the user, or the user has SPECIAL authority and SETROPTS SAUDIT is in effect.
 - g. An ACEE is to be created and a certificate has been specified that does not correspond to a RACF user ID.

- h. An ACEE is to be created and a certificate has been specified that is not trusted.
- 7. If an ACEE is to be anchored off the current TCB, then the INTA_TASK_LVL attribute must be set. Any value passed in ACEE_ptr is ignored, and the ACEE address is not returned. If an ACEE address is to be returned, the INTA_TASK_LVL attribute must be off. This results in the ACEE address being returned in the ACEE_ptr parameter area.
- 8. If an ACEE is to be deleted from the current TCB, then the INTA_TASK_LVL attribute must be set. If this is not done, the ACEE_ptr parameter must point to the address of the ACEE to be deleted.
- 9. If the function_code and attributes indicate that an ACEE is to be created and anchored off the TCB and there is an ACEE already anchored off the TCB, the caller receives a parameter list error.
- 10. If the function_code and attributes indicate that an ACEE is to be deleted from the TCB and there is no ACEE anchored to the TCB, the caller receives a parameter list error.
- 11. If the last word in the parameter list does not have a 1 in the high-order (sign) bit, the caller receives a parameter list error. The first parameter that can have the high-order bit on, ending the parameter list, is the logstring parameter
- 12. When the application is terminating and there are no tasks outstanding, initACEE should be called to purge all the managed ACEEs. Then the ACEE for the application server address space can be deleted.
- 13. The RACROUTE service should not be used to delete the managed ACEEs.
- 14. You can find parameter usages in Table 16 on page 41.
- 15. The service serializes resources at the address space level with a STEP ENQ on QNAME "SYSZRACF".
- 16. If the function_code indicates that an ACEE is to be created and the length of the certificate parameter is not zero, then the length of the RACF_user ID and password must both be 0. If a RACF_user ID or password is supplied with the certificate, the caller receives a parameter list error.
- 17. If the function_code indicates that an ACEE is to be deleted or that managed ACEEs should be purged, and the length of the certificate parameter is not zero, then the caller receives a parameter list error.
- 18. If the function_code indicates that a certificate is to be registered, deregistered, or queried, and the length of the certificate parameter is zero, then the caller receives a parameter list error.
- 19. The certificate supplied by the certificate parameter is used only to identify a RACF user ID. It is expected that the certificate was previously verified. Note the following additional details regarding initACEE's certificate processing:
 - a. All fields as defined for X.509 version 1 certificates must be present and non-null.
 - b. X.509 certificates with version numbers greater than 3 are not supported.
 - c. Version 3 certificates with critical extensions are not supported. Noncritical extensions are ignored.
 - d. Subject and issuer names can contain only the following string types:
 - T61STRING- TAG 20
 - PRINTABLESTRING- TAG 19
 - IA5STRING- TAG 22
 - VISIBLESTRING- TAG 26
 - GENERALSTRING- TAG 27

initACEE

- BMPString-TAG 30 (ASCII Unicode only)
- UTF8-TAG 12 (7 bit ASCII only)
- e. The length of the serial number plus the length of the issuer's name cannot exceed 245.
- f. No date validity check is performed on the certificate.
- g. No signature check is performed on the certificate.

If the `function_code` indicates that an ACEE is to be created, or that a certificate is to be queried, the certificate must be a single DER encoded X.509 certificate.

If the `function_code` indicates that a certificate is to be registered or deregistered, it must be in one of the following formats:

- a. A single DER encoded X.509 certificate.
- b. A Privacy Enhanced Mail (PEM) encoded X.509 certificate. If the input is in this format, only the Originator Certificate is used.
- c. One or more X.509 certificates contained within a PKCS #7 DER encoding. If the input is in this format, only the first certificate in the PKCS #7 encoding will be used.
- d. A Base64 encoded X.509 certificate as returned from a PKCS #10 certificate request. The data must include the string

```
'-----BEGIN CERTIFICATE-----'
```

immediately prior to the Base64 encoding, and the string

```
'-----END CERTIFICATE-----'
```

immediately following.

If transmitted from an ASCII system, PEM and Base64 encoded certificates must be translated from ASCII to EBCDIC before being passed to `initACEE`.

20. If the `function_code` indicates that a certificate is to be queried, the caller is expected to supply a 9 byte area for the `RACF_userid` parameter. If a user ID is associated with the certificate, `initACEE` updates this area with the length and value of the user ID.
21. If the `function_code` indicates that an ACEE is to be created or that a certificate is to be queried, and the certificate supplied by the caller is defined to RACF with a status of NOTRUST, `initACEE` will return a RACF return code 8 and a RACF reason code 40, indicating that no user ID is defined to use this certificate.
22. If the `function_code` and attributes indicate that an ACEE is to be created and an ENVR object is to be returned, then the `ENVR_out` parameter must point to a data structure for the ENVR object. The caller receives a parameter list error if the high order bit of a previous parameter indicates the end of the parameter list.
23. If the attributes indicate that an ENVR object is to be returned, it is the caller's responsibility to free the ENVR object storage. The caller should check the storage area length and address to determine if storage needs to be freed, not the `initACEE` return code. In some cases, an error may be encountered after creation of the ENVR object, resulting in a non-zero return code. The caller is still responsible for freeing the ENVR object in these cases.
24. If the `function_code` indicates that an ACEE is to be created, and the `ENVR_in` parameter points to an ENVR object data structure, then the length of the

RACF_userid, Password, and Certificate parameters must all be 0. The caller receives a parameter list error if a RACF_userid, Password, or Certificate is supplied with the ENVR_in parameter.

25. If the function_code indicates that an ACEE is to be deleted or that managed ACEEs should be purged, and the ENVR_in or ENVR_out parameter is specified, then the caller receives a parameter list error.
26. When an ENVR object is supplied with the ENVR_in parameter and an ACEE creation is requested, the attribute bits that affect the ACEE (INTA_UNAUTH_CLIENT, INTA_AUTH_CLIENT, INTA_NO_TIMEOUT) and the application name (APPL_id) are ignored.
27. An OUSP can only be returned if the Attributes parameter also indicates that a USP should be created (INTA_OUSP_RET should only be on if INTA_USP is on). If an OUSP is requested without a USP, then the caller receives a parameter list error.
28. If the Attributes parameter indicates that an OUSP should be returned (INTA_OUSP_RET), then the Output_area parameter must be specified. If it is not, the caller receives a parameter list error.
29. When the INTA_NO_TIMEOUT bit and the INTA_MANAGED bit are set for the creation of a new managed ACEE, the ACEE is cached and does not expire after five minutes.
30. If the Attributes parameter indicates that a no timeout ACEE is requested (INTA_NO_TIMEOUT) and a managed ACEE with an expiration time is found in the cache that satisfies the request, the address of the managed ACEE is returned. The expiration time of the ACEE in the cache remains the same. After receiving a subsequent delete request, the ACEE may expire.
31. If the function indicates that a certificate is being queried, and the Attributes parameter indicates that an X500 name pair should be returned (INTA_X500_RET), then both the DIGTCERT and DIGTNMAP profiles will be checked for a user ID associated with the certificate. If the function indicates that a certificate is being queried, and the X500 name pair has not been requested, then only the DIGTCERT profiles will be checked to determine if the certificate has been defined to RACF, and associated with a user ID.
32. If the Attributes parameter indicates that an X500 name pair should be returned (INTA_X500_RET), then the X500name parameter must be specified. If it is not, the caller receives a parameter list error.
33. If a certificate is being queried, and an X500 name pair has been requested, DIGTNMAP profiles may be used to determine the RACF user ID. If there are additional criteria associated with the DIGTNMAP profile, the APPL_id and Variable_list parameters, as well as the system-identifier of the system initACEE is running on, are used in determining the RACF user ID. If the certificate supplied for the query will later be used to create an ACEE, and the same user ID is expected to result, then the additional criteria must be the same for the query and create functions. In other words, the APPL_id and Variable_list parameter specifications must be the same, and the query and create must execute on the same system.
34. When the ENVR_in parameter is specified, the INTA_USP attribute is ignored. If the ENVR_in contains a USP, the resulting ACEE will also have a USP associated with it. The X500name parameter will also be ignored. If the ENVR_in contains an X500 name, the resulting ACEE will also have an X500 name associated with it. The information needed to create an OUSP is not available in an ENVR_object. If the INTA_RET_OUSP attribute is set indicating an OUSP should be returned, and an ENVR object is supplied with the ENVR_in parameter, the caller receives a parameter list error.

initACEE

35. If the `function_code` indicates that an ACEE is to be created or that a certificate is to be queried and processing determines that the user ID to be used is to be extracted from the `hostIdMappings` certificate extension, `InitACEE` will ignore the extension under the following conditions:
- The caller does not have READ authority (or greater) to any SERVAUTH class resource identified by the `hostName(s)` in the extension.
 - The user ID extracted has a length less than 1 or greater than 8.
 - For create only, the user ID is not a RACF defined user.
 - The definition of the `hostIdMappings` extension in ASN.1 syntax is:

```
id-ce-hostIdMappings OBJECT IDENTIFIER ::= { 1 3 18 0 2 18 1 }
```

```
HostIdMappings ::= SET OF HostIdMapping
```

```
HostIdMapping ::= SEQUENCE {  
    hostName          IMPLICIT[1] IA5String,  
    subjectId         IMPLICIT[2] IA5String,  
    proofOfIdPossession IdProof OPTIONAL  
}  
IdProof ::= SEQUENCE {  
    secret            OCTET STRING,  
    encryptionAlgorithm OBJECT IDENTIFIER  
}
```

36. If the `function_code` indicates that a certificate is to be registered, and one of the CERTAUTH certificates meets the following three conditions, the input certificate is treated as a certificate authority certificate to be registered as CERTAUTH:
- The CERTAUTH certificate's public key matches that of the input certificate.
 - The CERTAUTH certificate's subject distinguished name matches that of the input certificate and
 - The CERTAUTH certificate has a private key.

Otherwise, the input certificate is treated as an end-user certificate to be registered with the current user ID.

37. When a return code of 8 8 40 is received from an `initACEE Create`, other occurrences could be if the DIGTCERT, DIGTNMAP, or DIGTCRIT class has not been processed using SETROPTS RACLIST, or if SETROPTS RACLIST was used, but the class was not RACLIST REFRESHed after the certificate or mapping was added or altered. See *z/OS Security Server RACF Command Language Reference* for information on the RACDCERT and SETROPTS commands.

Related Services

None

initUSP (IRRSIU00): Initialize USP

Function

The `initUSP` service verifies that the user is authorized to use z/OS UNIX and, if so, establishes security attributes for the calling process. The `initUSP` service also returns any z/OS UNIX limits that have been set on a user basis.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

None

Format

```
CALL IRRSIU00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Output_area
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space and start on a word boundary.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Output_area

The name of a fullword in which the service routine stores the address of an area containing data about the user. IRRSIU00 uses the high-order bit of this fullword to determine if z/OS UNIX requested default processing for failures that occur under certain circumstances (as described in Note 1). If the high-order bit is on, an initUSP that would normally fail because of missing information completes successfully and builds a default USP. With current default processing (described for the getUMAP and getGMAP callable services and in usage notes for this service) even without the high-order bit turned on

initUSP

under the circumstances described in Notes 4 and 5, an initUSP that would previously fail is now completed successfully.

For all successful initUSP requests, the output area is obtained in the primary address space and must be freed by the caller of initUSP. The following data is returned:

- TSO/E user ID
- z/OS UNIX user identifier (UID) of user
- z/OS UNIX group identifier (GID) of current group
- Home directory path name
- Initial program path name
- User limits (when OUSP version is greater than 0)

The actual format of the output area is mapped by macro IRRPOUSP.

See *z/OS Security Server RACF Data Areas* for the actual format of the output area.

Return and Reason Codes

IRRSIU00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	12	An internal error occurred during RACF processing.
8	8	16	Recovery could not be established.
8	8	20	The user's profile has no OMVS segment.
8	8	24	The OMVS segment in the user's profile has no UID.
8	8	28	The OMVS segment in the current group's profile has no GID.

Usage Notes

1. This service is intended only for use by z/OS UNIX servers and the MVS BCP. It can be directly invoked by a z/OS UNIX server. The high-order bit of the output_area is set on by the caller when initUSP is called to establish the security attributes for critical z/OS UNIX address spaces such as the kernel. When the bit is on, initUSP builds a default z/OS UNIX security environment in certain cases when it would normally fail. InitUSP sets a SAF return code of 0, RACF return code of 0, RACF reason code of 0, and builds a default USP for the following cases:

- The user ID is not defined to RACF
- There is no OMVS segment in the user's profile
- There is no UID in the OMVS segment of the user's profile
- There is no GID in the OMVS segment of the current group's profile

The default USP returned to the caller (mapped by IRRPOUSP) contains a z/OS UNIX user identifier (UID) and z/OS UNIX group identifier (GID) of 0. The lengths for initial program and home directory path names is 0. If the user

ID is defined to RACF, the user ID is returned as an TSO/E user ID. If the user ID is not defined to RACF, the TSO/E user ID is set to an asterisk in the returned USP.

2. The address space or task must have an ACEE when this service is called.
3. A RACF user can be connected to more than NGROUPS_MAX groups, but only up to the first NGROUPS_MAX groups will be associated with the process when the process is created.

The first NGROUPS_MAX z/OS UNIX groups to which a user is connected (as shown by a LISTUSER command) get associated with the process.

4. If no OMVS segment is found in the user's profile, the initUSP service checks the BPX.DEFAULT.USER profile in the FACILITY class. This profile may contain a user ID in its application data field that provides a default OMVS segment. If this profile is found and the FACILITY class is active, it is used to set the home, PROGRAM, and user limits for the user.
5. If no OMVS segment is found in the group profile of the user's current connect group, the initUSP service checks the BPX.DEFAULT.USER profile in the FACILITY class. This profile may contain a group name in its application data field that provides a default OMVS segment. If this profile is found and the FACILITY class is active, it is used to set the GID for the user.

See *z/OS Security Server RACF Security Administrator's Guide* for information on APPLDATA in the FACILITY class.

Related Services

deleteUSP

makeFSP (IRRSMF00): Make IFSP

Function

The **makeFSP** service builds an IFSP in the area provided by the caller.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

None

Format

```
CALL IRRSMF00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Mode,
               ALET, Output_FSP,
               ALET, Owing_directory_FSP,
               ALET, File_Identifier,
               ALET, CRED
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Mode

The name of a word containing the mode values (the filetype, the permission bits, and the S_ISUID, S_ISGID, and S_ISVTX bits) to be set for the file.

See "File Type and File Mode Values" on page 4 for a definition of the security bits in the mode parameter.

Output_FSP

The name of a 64-byte area in which the new IFSP is built.

Owing_directory_FSP

The name of an area containing the IFSP for the owing directory.

File_Identifier

The name of a 16-byte area containing a unique identifier of the file.

CRED

The name of the CRED structure for the current file system syscall. See *z/OS Security Server RACF Data Areas*.

Return and Reason Codes

IRRSMF00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
8	8	12	An internal error occurred during RACF processing.
8	8	32	CRED user type is not supported.
8	12	4	A model Access Control List (ACL) exists for the parent, but no buffer address was provided for the new object's access ACL in the CredAccAcl field.
8	12	8	The buffer provided (in the CredAccAclLen field) for the new object's access ACL is not large enough. It must be at least as large as the size of the parent's directory model ACL or file model ACL (in the FACL_Len field), as appropriate to the type of object being created.
8	12	12	A directory model ACL exists for the parent, but no buffer address was provided for the new directory's directory model ACL in the CredDirModelAcl field
8	12	16	The buffer provided (in the CredDirModelAclLen field) for the new directory's directory model ACL is not large enough. It must be at least as large as the size of the parent's directory model ACL (in the FACL_Len field)
8	12	20	A file model ACL exists for the parent, but no buffer address was provided for the new directory's file model ACL in the CredFileModelAcl field
8	12	24	The buffer provided (in the CredFileModelAclLen field) for the new directory's directory model ACL is not large enough. It must be at least as large as the size of the parent's directory model ACL (in the FACL_Len field)

Usage Notes

1. This service is only intended for use by a z/OS UNIX file system and by z/OS UNIX servers. The service contains support for z/OS UNIX servers, but cannot be directly invoked by a z/OS UNIX server.
2. If the CRED user type is system, IRRSMF00 allows the operation, and sets the owning z/OS UNIX user identifier (UID) to zero.
3. IRRSMF00 builds the IFSP in the output_FSP area provided by the caller. The caller must save the IFSP as part of the attributes for the object.
4. IRRSMF00 builds the IFSP with the S_ISUID bit set to zero and the S_ISVTX bit set to the value in the mode byte. If the new object is a directory, and the FILE.GROUPOWNER.SETGID profile exists in the UNIXPRIV class, the S_ISGID bit is inherited from the parent directory. Otherwise, the S_ISGID bit is set to zero.
5. The new object's owning UID is set to the effective UID of the process. By default, the owning GID is set to that of the parent directory. However, if the

makeFSP

FILE.GROUPOWNER.SETGID profile exists in the UNIXPRIV class, then the owning GID is determined by the set-gid bit of the parent directory as follows:

- If the parent's set-gid bit is on, then the owning GID is set to that of the parent directory.
 - If the parent's set-gid bit is off, then the owning GID is set to the effective GID of the process.
6. If the parent directory has a directory model ACL, and the new object is a directory, then the parent's directory model ACL is copied as the new directory's access ACL and directory model ACL. The caller must pass in the address of the parent's directory model ACL in the CredPDirModelAcl field. The caller must pass in the length and address of buffers to contain both the new directory's access ACL and directory model ACL. The buffers must be large enough to contain the copied ACL. The address of the new directory's directory model ACL buffer must be passed in using the CredDirModelAcl field, and its length must be passed in using the CredDirModelAclLen field. The address of the new directory's access ACL buffer must be passed in using the CredAccAcl field, and its length must be passed in using the CredAccAclLen field.
 7. If the parent directory has a file model ACL, and the new object is a directory, then the parent's file model ACL is copied as the new directory's file model ACL. The caller must pass in the address of the parent's file model ACL in the CredPFileModelAcl field. The caller must pass in the length and address of a buffer to contain the new directory's file model ACL. The buffer must be large enough to contain the copied ACL. The address of the new directory's file model ACL buffer must be passed in using the CredFileModelAcl field, and its length must be passed in using the CredFileModelAclLen field.
 8. If the parent directory has a file model ACL, and the new object is a file, then the parent's file model ACL is copied as the new file's access ACL. The caller must pass in the address of the parent's file model ACL in the CredPFileModelAcl field. The caller must pass in the length and address of a buffer to contain the new file's access ACL. The buffer must be large enough to contain the copied ACL. The address of the new file's access ACL buffer must be passed in using the CredAccAcl field, and its length must be passed in using the CredAccAclLen field.

Related Services

ck_access, R_umask

makeISP (IRRSMI00): Make IISP

Function

The **makeISP** service builds an IISP in the area provided by the caller.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts

Locks: No locks held

Control parameters: The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

None

Format

```
CALL IRRSMI00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Mode_Permissions,
               ALET, Output_ISP,
               ALET, Output_IPCP,
               ALET, CREDIPC
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Mode_Permissions

The name of a word containing the mode permission flags to be set for this IPC key. The following is a list of defined permission bits mapped by BPXYMODE:

S_IRUSR

Permits the process that owns the IPC member to read it.

S_IWUSR

Permits the process that owns the IPC member to alter it.

S_IRGRP

Permits the group associated with the IPC member to read it.

S_IWGRP

Permits the group associated with the IPC member to alter it.

S_IROTH

Permits others to read the IPC member.

makeISP

S_IWOTH

Permits others to alter the IPC member.

Alter and write have the same meaning for access checks. Alter applies to semaphores and write applies to message queueing and shared memory segments.

Output_ISP

The name of a 64-byte area in which the new IISP is built. The name is set by the kernel. See *z/OS Security Server RACF Data Areas*.

Output_IPCP

The name of a 20-byte area in which the new IPCP is built. The name is set by the kernel.

CREDIPC

The name of the CREI structure for the current IPC system callable service. The CREI contains the IPC identifier and IPC key. See *z/OS Security Server RACF Data Areas*.

Return and reason codes

IRRSMI00 may return the following values in the reason and return code parameters:

SAF return code	RACF return code	RACF reason code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	12	An internal error occurred during RACF processing.
8	8	32	The CREI user type is not supported.

Usage Notes

1. This service is only intended for use by the MVS BCP.
2. The CREI user type must be local (that is, 1).
3. IRRSMI00 builds the IISP in the output_ISP area and the output_IPCP areas provided by the caller. The caller must save the IISP as part of the attributes for the key.
4. The IPCP ALET and address are retrieved from the parameters and set into the output_ISP by RACF.
5. The effective z/OS UNIX user identifier (UID) and z/OS UNIX group identifier (GID) are retrieved from the USP and set into the owner and creator fields of the output_IPCP by RACF.
6. The mode is retrieved from the parameters and set into the output_IPCP by RACF.
7. The IPC Key and IPC ID are retrieved from the CREI and set into the output_ISP by RACF.
8. An audit record is optionally written, depending on the audit options in effect for the system.
9. This service uses task level support when z/OS UNIX has indicated in the task's ACEE that this is a task level process.

Related services

ck_IPC_access, R_IPC_ctl

make_root_FSP (IRRSMR00): Make Root IFSP

Function

The **make_root_FSP** service initializes an IFSP for the root directory of a new file system being initialized in a hierarchical file system (HFS) data set.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

None

Format

```
CALL IRRSMR00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Mode,
               ALET, Output_FSP,
               ALET, File_Identifier,
               ALET, Data_set_name
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

make_root_FSP

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Mode

The name of a word containing the mode value (the file type, the permission bits, and the S_ISUID, S_ISGID, and S_ISVTX bits) to be set for the file.

See “File Type and File Mode Values” on page 4 for a definition of the security bits in the mode parameter.

Output_FSP

The name of a 64-byte area in which the new IFSP is built.

File_Identifier

The name of a 16-byte area containing a unique identifier of the root directory.

Data_set_name

The name of an area containing the data set name of the HFS data set being created. This is a 44-byte area padded with blanks.

Return and Reason Codes

IRRSMR00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	12	An internal error occurred during RACF processing.
8	8	16	Recovery could not be established.

Usage Notes

1. This service is only intended for use by DFSMS/MVS, during allocation of an HFS data set, and by z/OS UNIX System Services servers. The service contains support for z/OS UNIX System Services servers, but can not be directly invoked by an z/OS UNIX System Services server.
2. IRRSMR00 may be called from a non-z/OS UNIX address space.
3. These are the default attributes set for the root directory:
 - The file’s owner z/OS UNIX user identifier (UID) and z/OS UNIX group identifier (GID) are initialized as follows:
 - If the caller is a z/OS UNIX process:
 - owner UID = effective UID of the process
 - owner GID = effective GID of the process

Note: This differs from **makeFSP** because there is no owning directory to propagate the GID from.

- If the caller is not a z/OS UNIX process but is defined to RACF as a z/OS UNIX user:
 - owner UID = UID from the user’s profile
 - owner GID = GID from the group profile for the user’s current groupIf the group has no GID, the owner GID is set to 0.

- If the caller is not a z/OS UNIX process and is not defined to RACF as a z/OS UNIX user:
 - owning UID = 0
 - owning GID = 0
 If the UID or GID is set to 0, a superuser should change the fields to valid values using **chown** after the file system is mounted.
 - The permission bits are set from the input mode parameter.
 - The S_ISUID and S_ISGID are set to 0 and the S_ISVTX bit is set to the value in the input mode parameter.
 - The user audit options are set to audit access failures for all types of access.
 - The auditor audit options are set to no auditing.
4. IRRSMR00 builds the IFSP in the output_FSP area provided by the caller. The caller must save the IFSP as part of the attributes for the object.

Related Services

makeFSP

query_file_security_options (IRRSQF00): Query File Security Options

Function

The **query_file_security_options** service returns the value of the requested file system option.

For a list of supported options, see the Option_code parameter.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	None
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

None

query_file_security_options

Format

```
CALL IRRSQF00 (Work_area,  
              ALET, SAF_return_code,  
              ALET, RACF_return_code,  
              ALET, RACF_reason_code,  
              ALET, Option_code,  
              ALET, Output_value  
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Option_code

The name of a word containing a code identifying the requested option. The code value 1 identifies the `_POSIX_CHOWN_RESTRICTED` option. Option code value 2 is sent as input to IRRSQF00 to determine if the system supports Access Control Lists (ACLs) or not.

Output_value

The name of a word in which the value of the requested option is returned.

- Option_code value 1

For option code value 1, the following may be returned:

0 `_POSIX_CHOWN_RESTRICTED` is in effect

-1 `_POSIX_CHOWN_RESTRICTED` is not in effect

Note:

If all of the following are true, then `_POSIX_CHOWN_RESTRICTED` is not in effect:

- The UNIXPRIV class is active
- The UNIXPRIV class has been processed using SETROPTS RACLIST
- The following discrete profile exists in the UNIXPRIV class: `CHOWN.UNRESTRICTED`.

- Option_code value 2

For option code value 2, the following may be returned:

0 ACLs are supported

-1 ACLs are not supported

RACF will always return 0, indicating that ACLs are supported.

Return and Reason Codes

IRRSQF00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The option code is not supported.

Usage Note

1. This service is intended only for use by a z/OS UNIX file system.

Related Services

None

query_system_security_options (IRRSQS00): Query System Security Options

Function

The **query_system_security_options** service returns the value of the requested system options. The only supported options are NGROUPS_MAX and _POSIX_SAVED_IDS.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	None
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. RACF returns the following values for the requested system options:
 - NGROUPS_MAX: 300
 - _POSIX_SAVED_IDS: 0

query_system_security_options

Format

```
CALL IRRSQS00 (Work_area,  
              ALET, SAF_return_code,  
              ALET, RACF_return_code,  
              ALET, RACF_reason_code,  
              ALET, Option_code,  
              ALET, Output_value  
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Option_code

The name of a word containing a code identifying the requested option. The supported values are:

- 1 NGROUPS_MAX
- 2 _POSIX_SAVED_IDS

All other values are reserved.

Output_value

The name of a word in which the value of the requested option is returned.

- The values for _POSIX_SAVED_IDS are:
 - 0 _POSIX_SAVED_IDS is in effect.
 - 1 _POSIX_SAVED_IDS is not in effect.
- The value for NGROUPS_MAX is the maximum number of supplemental groups supported.

Return and Reason Codes

IRRSQS00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The option code is not supported.

Usage Note

- This service is intended only for use by the MVS BCP.

Related Services

None

R_admin (IRRSEQ00): RACF Administration API

Function

The **R_admin** service enables applications to construct a function code/data field name parameter list to manage RACF profiles within the RACF database. Alternately, a RACF TSO administrative command image can be passed to manage RACF user profile information. Output, if any, which resulted from RACF's processing of the profile admin request is returned to the caller in virtual storage. This callable service does **not** support all RACF command functions. For a list of the commands that are **not** executed through this service, see the Usage Notes section.

The exact format (spacing and order) of the data in the command output or messages does not form a programming interface. No programs should depend on the exact format of this data.

Requirements

Authorization:	Any PSW key in supervisor or problem state
Dispatchable unit mode:	Any task
Cross memory mode:	PASN = HASN = SASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary only
Recovery mode:	Recovery must be provided by caller
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space.

RACF authorization

For the function codes which cause a RACF command to execute in the RACF address space, the command will run under the authority of a RACF user ID. The RACF user ID can come from one of a number of sources, and is searched for in the following order:

- The RACF_userID parameter
- The ACEE_ptr parameter
- The user ID associated with the current task control block (TCB)
- The user ID associated with the current address space (ASXB)

For problem state callers only, READ authority to the resource IRR.RADMIN.(*command-name*) in the FACILITY class is required to execute the RACF TSO command *command-name* using R_Admin. This is in addition to any authority checks done by the command itself. The resource must be defined using the full command name even if the abbreviated version of the command name is

R_admin

used with R_Admin. (e.g., lu joeuser would require READ authority to IRR.RADMIN.LISTUSER.) Generic profiles can be used.

Format

```
CALL IRRSEQ00 (Work_area,  
              ALET, SAF_return_code,  
              ALET, RACF_return_code,  
              ALET, RACF_reason_code,  
              Function_code,  
              Parm_list,  
              RACF_userID,  
              ACEE_ptr,  
              Out_message_subpool,  
              Out_message_strings  
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The word containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code. These return codes are found in Table 21 on page 65.

RACF_return_code

The name of a fullword in which the service routine stores the return code. These return codes are found in Table 21 on page 65.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code. These reason codes are found in Table 21 on page 65.

Function_code

The name of a one-byte field that specifies the function (administration request) that RACF is to perform. The function code may have one of the values found in Table 17 on page 63.

Parm_list

The name of a variable marking the start of the input parameter list. The mapping macro IRRPCOMP contains a definition of the parameter list for each of the values of function_code. To find parameter list mappings for the values of function_code, see Table 18 on page 64.

RACF_userID

The name of a 9 byte area that consists of a 1 byte length field followed by the userID, which can be up to eight characters. If not specified, the length must equal zero. Otherwise, the user ID must be specified in upper case. If specified, the RACF command executed through R_admin will run under the authority of this userID.

ACEE_ptr

The name of a fullword containing the address of the ACEE of the user under whose identity the RACF administrative request runs. The user ID is extracted

from the ACEEUSER field. The ACEE itself is not used for subsequent authority checking for the request. If the caller does not specify an ACEE, this area must contain binary zeros. If both an ACEE and a user ID are passed into this service, the user ID is used.

Out_message_subpool

The name of a one-byte field that specifies the subpool used to obtain storage for output messages that are returned. Problem state callers are limited to subpools 1 thru 127.

Out_message_strings

The name of a fullword in which the service routine stores the address of output data, if applicable. It is the responsibility of the caller to free the output storage.

For SETROPTS functions (ADMN_XTR_SETR and ADMN_UNL_SETR), see Table 67 on page 98 for the output mapping.

For the remaining functions, see Table 19 on page 64 for the mapping of the output message block returned by this service. For the format of each message entry, see Table 20 on page 65. The message entry is a repeating data structure. In it, the entry length and message text repeats if more than one message is present. See also the IRRPCOMP data area in *z/OS Security Server RACF Data Areas*.

Function code values

Table 17 shows the function code values in the mapping macro IRRPCOMP.

Table 17. Function Code Values in Mapping Macro IRRPCOMP

Function Code	Value	Description
ADMN_ADD_USER	X'01'	Add a user to the RACF database
ADMN_DEL_USER	X'02'	Delete a user from the RACF database
ADMN_ALT_USER	X'03'	Alter a user's RACF database profile
ADMN_LST_USER	X'04'	List the contents of a user's RACF database profile
ADMN_RUN_COMD	X'05'	Run a RACF command image
ADMN_ADD_GROUP	X'06'	Add a group to the RACF database
ADMN_DEL_GROUP	X'07'	Delete a group from the RACF database
ADMN_ALT_GROUP	X'08'	Alter a group's RACF database profile
ADMN_LST_GROUP	X'09'	List a group's RACF database profile
ADMN_CONNECT	X'0A'	Connect a single user to a RACF group
ADMN_REMOVE	X'0B'	Remove a single user from a RACF group
ADMN_ADD_GENRES	X'0C'	Add a general resource profile to the RACF database
ADMN_DEL_GENRES	X'0D'	Delete a general resource profile from the RACF database
ADMN_ALT_GENRES	X'0E'	Alter a general resource's RACF database profile
ADMN_LST_GENRES	X'0F'	List a general resource's RACF database profile
ADMN_ADD_DS	X'10'	Add a data set profile to the RACF database
ADMN_DEL_DS	X'11'	Delete a data set profile from the RACF database
ADMN_ALT_DS	X'12'	Alter a data set's RACF database profile
ADMN_LST_DS	X'13'	List a data set's RACF database profile

Table 17. Function Code Values in Mapping Macro IRRPCOMP (continued)

Function Code	Value	Description
ADMN_PERMIT	X'14'	Permit a user or group to a RACF profile
ADMN_ALT_SETTR	X'15'	Alter SETROPTS information
ADMN_XTR_SETTR	X'16'	Extract SETROPTS information in R_admin format
ADMN_UNL_SETTR	X'17'	Extract SETROPTS information in SMF data unload format

Parameter list mappings by function code

Table 18 shows where to find the parameter list mappings for the values of function_code.

Table 18. Parameter List Mappings for Function_Code Values

For function_code(s)	See
ADMN_ADD_USER, ADMN_DEL_USER, ADMN_ALT_USER, ADMN_LST_USER	"User administration" on page 68
ADMN_RUN_COMD	"Running RACF commands" on page 68
ADMN_ADD_GROUP, ADMN_DEL_GROUP, ADMN_ALT_GROUP, ADMN_LST_GROUP	"Group administration" on page 77
ADMN_CONNECT ADMN_REMOVE	"Group connection administration" on page 80
ADMN_ADD_GENRES, ADMN_DEL_GENRES, ADMN_ALT_GENRES, ADMN_LST_GENRES, ADMN_ADD_DS, ADMN_DEL_DS, ADMN_ALT_DS, ADMN_LST_DS, ADMN_PERMIT	"Resource profile administration" on page 82
ADMN_ALT_SETTR	"Parameter list format for SETROPTS administration" on page 92
ADMN_XTR_SETTR	Input parameter list is ignored
ADMN_UNL_SETTR	Input parameter list is ignored

Output message block mapping

Table 19. Mapping of Output Message Block

Offset	Length	Description
0	4	Next ADMIN output messages block, or zero if no additional blocks follow
4	4	Eye catcher to aid in virtual storage dumps 'RMSG'
8	1	Storage subpool in which the block was obtained
9	3	Total block length
12	4	Offset to the first byte after the last message; This offset value is related to the first message.
16	1	Start of the first message

Table 20. Format of Each Message Entry

Offset	Length	Description
0	2	Length of this message text entry.
2	*	Variable message text.

The actual format of the output area is mapped by macro IRRPCOMP.

See *z/OS Security Server RACF Data Areas* for the actual format of the output area.

Return and reason codes

IRRSEQ00 returns the following values in the reason and return code parameters:

Table 21. Return and Reason Codes

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful. Output from the RACF command may be present. The <code>Out_message_strings</code> parameter should be interrogated by caller.
4	0	0	RACF is not installed.
8	8	0	Incorrect function code
8	8	4	Input parameter list error
8	8	8	Invalid data in the input parameter list may cause the program to abend. Some fields in the parameter list must be coded with the defined length and data format. The profile segment name, for example must not be longer than 8 bytes. If more than 8 bytes are passed, unpredictable results may occur and the user gets a return code of 8 8 8 with a possible program abend. If the program ABENDs before GTF trace records are created, then you do not have GTF trace records to use for debugging. Check the input parameter list for errors.
8	8	12	Recovery environment could not be established.
8	8	16	Invalid request specified. For <code>ADMN_RUN_COMD</code> , the input command image represented an incorrect or unsupported command. For all other functions, the input parameter list contained an incorrect segment name, field name, or flag byte, or incorrectly specified field data. The function-specific parameter list header contains an offset to the segment or field entry in error, relative to the start of the parameter list.
8	8	20	Function not supported for problem state caller.

Table 21. Return and Reason Codes (continued)

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
8	8	24	Problem state caller not authorized to issue command.
8	12	<i>IEFSSREQ return code</i>	Unable to invoke the RACF subsystem. Reason code field contains a return code from the IEFSSREQ macro invocation. See <i>z/OS MVS Using the Subsystem Interface</i> for information on possible return codes from IEFSSREQ.
8	16	<i>RACF command return code</i>	RACF request failed. Reason code field contains the return code from the RACF request. See <i>z/OS Security Server RACF Command Language Reference</i> for information on possible return codes from the RACF commands. In addition, diagnostic output resulting from the RACF command may be present. The <code>Out_message_strings</code> parameter should be interrogated by the caller.

Note: Return and reason codes are shown in decimal.

Usage notes

1. You must link edit the IRRSEQ00 callable service stub into your application code to resolve the entry point address at run time.
2. For the `Out_message_subpool` parameter, select a subpool carefully. z/OS makes certain assumptions about subpool usage and characteristics. Using subpool 0 or 250 or any subpool documented in *z/OS Application Development Guide* as having a storage key of USER (for example, 227-231 and 241) may give unpredictable results.
3. The RACF subsystem must be active to use this service.
4. The caller of this service must free the message output blocks returned by this service.
5. All requests are processed synchronously. Control is not returned to the caller until RACF has processed the administration request and output, if any has been returned to the caller.
6. For the `ADMN_RUN_COMD` function code, the following RACF commands are not supported through this interface:
 - BLKUPD
 - RACLINK
 - RVARY
 - RACF operator commands (DISPLAY, RESTART, SET, SIGNOFF, STOP, and TARGET)

RACF TSO administrative commands may not be directed to other RACF remote sharing facility (RRSF) nodes. The command image passed by the caller cannot contain the keywords AT or ONLYAT. These keywords cause the command to fail with SAF return code 8, RACF return code 16, RACF reason code 8.

These messages are returned as command output:

```
IRRV013I subsystem-name SUBSYSTEM racf-command COMMAND FROM
THE IRRSEQ00 CALLABLE SERVICE WAS NOT PROCESSED.
```

```
IRRV014I subsystem-name SUBSYSTEM AT() OR ONLYAT() KEYWORDS
MAY NOT BE SPECIFIED WITH COMMANDS FROM THE IRRSEQ00
CALLABLE SERVICE.
```

7. The command text must be left-justified within the input buffer
8. If a RACF command runs and does not return any output, the `Out_message_strings` parameter will be zero.
9. The parameter list passed to this service is a variable-length (VL) parameter list. The high-order bit of the last field (address of `Out_message_strings`) must be set to mark the end of the parameter list.
10. All field data must be supplied in character format. For information about the contents of the field data, refer to *z/OS Security Server RACF Command Language Reference* for the appropriate command keyword as indicated in the following tables. For example, looking at Table 28 on page 71 to find details on the content of the HLDCLASS field, see the ADDUSER/ALTUSER documentation for the HOLDCLASS keyword of the TSO segment.
 Additionally, RACF has a restriction of no more than 255 operands affecting a single nonbase segment (such as the TSO segment in a user profile, or the TME segment in a general resource profile) on a single command. Since the R_admin callable service generates a RACF command, this restriction applies to the number of field operands affecting nonbase segments. See the "RACF command restriction for nonbase segments in RACF profiles" section in the *z/OS Security Server RACF Command Language Reference* for specifics on that restriction.
11. For the list functions, the output for each segment is returned in the order in which the segments were supplied, with the exception that the BASE segment information is always returned first.
12. Any update to the RACF database caused by this service is subject to automatic direction and password synchronization as implemented by the installation.
13. The amount of output returned from any command run by this service is subject to the limits established for RRSF.
14. The following errors result in a "input parameter list error" being returned to the caller:
 - VL bit not set
 - An incorrectly specified `ADMN_USRADM_USER_LEN`, `ADMN_GRPADM_LEN`, or `ADMN_RESADM_CLAS_LEN` (must be from 1-8, inclusively)
 - An incorrectly specified length for the RACF user ID parameter (must be from 0 to 8, inclusively)
 - Setting `ADMN_USRADM_SEGM_NUM=0` on any of the list functions
 - Omitting the PROFILE field on any of the general resource, data set (except list) or permit function codes
 - Specifying a subpool outside the range of 1 to 127 when the caller is in problem state
15. The only supported function for problem state callers is `ADMIN_RUN_COMD`. The `RACF_userid` and `ACEE_ptr` parameters are ignored for problem state callers.

Related services

None

Parameter list formats

The following information describes parameter list formats for running RACF commands and for all administration.

Running RACF commands

For the ADMN_RUN_COMD (execute a RACF command) function code, the mapping associated with the function-specific parameter list is mapped as follows:

Table 22. Parameter List Format for Running a Command

Offset	Length	Description
0	2	Length of the RACF command string. Note, the length must not exceed 4096 characters.
2	*	Syntactically correct RACF TSO administration command string.

User administration

For the ADMN_ADD_USER, ADMN_DEL_USR, ADMN_ALT_USER, and ADMN_LST_USER function codes, the mapping associated with the function specific parameter list is mapped as in Table 23.

Table 23. Parameter List Format for User Administration

Offset	Length	Description
0	1	Length of the user ID
1	8	Uppercase RACF user ID
9	1	Reserved
10	2	Output offset to the segment or field entry in error in relationship to the start of the Parm_list. Only applies to ADMN_ADD_USER and ADMN_ALT_USER requests when an "invalid request" error is returned to the caller.
12	2	Number of RACF profile segments
14	*	Start of first segment entry

Note: For ADMN_DEL_USER, no segment data is expected. The number of segments should be zero. If non-zero, any segment data present is ignored.

Each segment entry is mapped as in Table 24.

Table 24. Segment Entry Mapping

Offset	Length	Description
0	8	Profile segment name (left-justified, uppercase, and padded with blanks). For ADMN_LST_USER, any user profile segment as defined in the RACF database templates is accepted. For ADMN_ADD_USER and ADMN_ALT_USER, the following segments are supported: BASE, CICS, DCE, DFP, KERB, LANGUAGE, LNOTES, NDS, NETVIEW, OMVS, OPERPARM, OVM, TSO and WORKATTR.

Table 24. Segment Entry Mapping (continued)

Offset	Length	Description
8	1	Flag byte. Use Y to create or alter the segment. Use N to delete the segment.
9	2	Number of fields to update within a segment
11	*	First field for segment

For ADMN_LST_USER, the flag byte is ignored.

For ADMN_ADD_USER, the flag byte indicates whether the segment should be created (Y) or should no longer exist (N). Because the BASE segment cannot be deleted, the flag byte for BASE is ignored. In addition, the flag byte of 'Y' may be specified for any segment name. When the callable service finds the 'Y' flag byte, it ignores the segment.

For ADMN_LST_USER, ADMN_ADD_USER, ADMN_ALT_USER with the flag byte N, no field data is expected. The number of fields should be zero. If non-zero, any field data present is ignored.

Each field entry is mapped as in Table 25.

Table 25. Field Entry Mapping

Offset	Length	Description
0	8	Field name as defined in the tables that follow. All field names must be left-justified, entered in all uppercase, and padded with blanks.
8	1	Field-specific flag byte
9	2	Length of field data
11	*	Field data

In all of the tables that describe ADDUSER and ALTUSER, the following rules for ADMN_ADD_USER and ADMN_ALT_USER apply:

- For boolean fields, the flag byte value of 'Y' or 'N' determines the flag setting. No field data is allowed for the boolean fields. Coding field data results in an "Invalid Request" error being returned to the caller.
- For text fields, the flag byte 'Y' indicates the field should be created (or altered) with the specified field data. The flag byte 'N' indicates the field should be deleted (or not created). For flag byte 'N', any field data specified is ignored.
- For repeating text fields, a flag byte of 'A' indicates to add the specified field data to the existing data (if any). Field data must be specified. If field data is not specified, an "Invalid Request" error is returned to the caller. The flag byte 'D' indicates to delete the specified data (if any).

In addition to the flag bytes specified in the following tables, the flag byte of 'Y' may be specified for any segment name. When the callable service finds the 'Y' flag byte, it ignores the segment.

The following tables define the field keywords and their usage. All field names relate directly to the ADDUSER and ALTUSER keywords. See *z/OS Security Server RACF Command Language Reference* for questions pertaining to field usage and data.

R_admin

Table 26. BASE Segment Fields

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
OWNER	'Y'	OWNER(xx)	Yes	Yes
ADSP (Boolean)	'Y'	ADSP	Yes	Yes
	'N'	NOADSP	Yes	Yes
SPECIAL (Boolean)	'Y'	SPECIAL	Yes	Yes
	'N'	NOSPECIAL	Yes	Yes
OPER (Boolean)	'Y'	OPERATIONS	Yes	Yes
	'N'	NOOPERATIONS	Yes	Yes
GRPACC (Boolean)	'Y'	GRPACC	Yes	Yes
	'N'	NOGRPACC	Yes	Yes
PASSWORD	'Y'	PASSWORD (xx)	Yes	Yes
	'N'	NOPASSWORD	Yes	Yes
EXPIRED (Boolean)	'Y'	EXPIRED	No	Yes
	'N'	NOEXPIRED	No	Yes
NAME	'Y'	NAME (xx)	Yes	Yes
	'N'	NAME	No	Yes
DFLTGRP	'Y'	DFLTGRP (xx)	Yes	Yes
GROUP	'Y'	GROUP (xx)	No	Yes
AUTH	'Y'	AUTHORITY (xx)	Yes	Yes
UACC	'Y'	UACC (xx)	Yes	Yes
MODEL	'Y'	MODEL (xx)	Yes	Yes
	'N'	NOMODEL	No	Yes
DATA	'Y'	DATA (xx)	Yes	Yes
	'N'	NODATA	No	Yes
UAUDIT (Boolean)	'Y'	UAUDIT	No	Yes
	'N'	NOUAUDIT	No	Yes
AUDITOR (Boolean)	'Y'	AUDITOR	Yes	Yes
	'N'	NOAUDITOR	Yes	Yes
OIDCARD (Boolean)	'Y'	OIDCARD	No	No
	'N'	NOOIDCARD	Yes	Yes
SECLEVEL	'Y'	SECLEVEL (xx)	Yes	Yes
	'N'	NOSECLEVEL	No	Yes
SECLABEL	'Y'	SECLABEL (xx)	Yes	Yes
	'N'	NOSECLABEL	No	Yes
CATEGORY	'Y'	ADDCATEGORY(xx ...)	Yes	No
	'A'	ADDCATEGORY(xx ...)	No	Yes
	'D'	DELCATEGORY(xx ...)	No	Yes
NOTE: To remove unknown categories from the profile, specify the 'D' flag and a field length of zero.				
REVOKE	'Y'	REVOKE(xx)	No	Yes
RESUME	'Y'	RESUME(xx)	No	Yes

Table 26. BASE Segment Fields (continued)

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
WHENDAYS	'Y'	WHEN(DAYS (xx))	Yes	Yes
WHENTIME	'Y'	WHEN(TIME (xx))	Yes	Yes
CLAUTH	'Y'	CLAUTH(xx...)	Yes	No
	'A'	CLAUTH(xx ...)	No	Yes
	'D'	NOCLAUTH(xx ...)	No	Yes
	'N'	NOCLAUTH	Yes	No
REST	'Y'	RESTRICTED	Yes	Yes
	'N'	NORESTRICTED	Yes	Yes

Table 27. OMVS Segment Fields

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
z/OS UNIX user identifier (UID)	'Y'	OMVS(UID (xx))	Yes	Yes
	'N'	OMVS(NOUID)	No	Yes
SHARED	'Y'	OMVS(SHARED)	Yes	Yes
AUTOID	'Y'	OMVS(AUTOUID)	Yes	Yes
HOME	'Y'	OMVS(HOME (xx))	Yes	Yes
	'N'	OMVS(NOHOME)	No	Yes
PROGRAM	'Y'	OMVS(PROGRAM (xx))	Yes	Yes
	'N'	OMVS(NOPROGRAM)	No	Yes
CPUTIMEMAX	'Y'	OMVS(CPUTIMEMAX (xx))	Yes	Yes
	'N'	OMVS(NOCPUTIMEMAX)	No	Yes
ASSIZEMAX	'Y'	OMVS(ASSIZEMAX (xx))	Yes	Yes
	'N'	OMVS(NOASSIZEMAX)	No	Yes
FILEPROCMAX	'Y'	OMVS(FILEPROCMAX (xx))	Yes	Yes
	'N'	OMVS(NOFILEPROCMAX)	No	Yes
PROCUSERMAX	'Y'	OMVS(PROCUSERMAX (xx))	Yes	Yes
	'N'	OMVS(NOPROCUSERMAX)	No	Yes
THREADSMAX	'Y'	OMVS(THREADSMAX (xx))	Yes	Yes
	'N'	OMVS(NOTHREADSMAX)	No	Yes
MMAPAREAMAX	'Y'	OMVS(MMAPAREAMAX (xx))	Yes	Yes
	'N'	OMVS(NOMMAPAREAMAX)	No	Yes

Table 28. TSO Segment Fields

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
ACCTNUM	'Y'	TSO(ACCTNUM (xx))	Yes	Yes
	'N'	TSO (NOACCTNUM)	No	Yes

Table 28. TSO Segment Fields (continued)

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
DEST	'Y'	TSO(DEST (xx))	Yes	Yes
	'N'	TSO(NODEST)	No	Yes
HLDCLASS	'Y'	TSO(HOLDCLASS (xx))	Yes	Yes
	'N'	TSO(NOHOLDCLASS)	No	Yes
JOBCLASS	'Y'	TSO(JOBCLASS (xx))	Yes	Yes
	'N'	TSO(NOJOBCLASS)	No	Yes
MSGCLASS	'Y'	TSO(MSGCLASS (xx))	Yes	Yes
	'N'	TSO(NOMSGCLASS)	No	Yes
PROC	'Y'	TSO(PROC(xx))	Yes	Yes
	'N'	TSO(NOPROC)	No	Yes
SIZE	'Y'	TSO(SIZE (xx))	Yes	Yes
	'N'	TSO(NOSIZE)	No	Yes
MAXSIZE	'Y'	TSO(MAXSIZE (xx))	Yes	Yes
	'N'	TSO(NOMAXSIZE)	No	Yes
SYSOUTCL	'Y'	TSO(SYSOUTCL (xx))	Yes	Yes
	'N'	TSO(NOSYSOUTCL)	No	Yes
USERDATA	'Y'	TSO(USERDATA (xx))	Yes	Yes
	'N'	TSO(NOUSERDATA)	No	Yes
UNIT	'Y'	TSO(UNIT (xx))	Yes	Yes
	'N'	TSO(NOUNIT)	No	Yes
COMMAND	'Y'	TSO(COMMAND (xx))	Yes	Yes
	'N'	TSO(NOCOMMAND)	No	Yes
SECLABEL	'Y'	TSO(SECLABEL (xx))	Yes	Yes
	'N'	TSO(NOSSECLABEL)	No	Yes

Table 29. CICS Segment Fields

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
OPIDENT	'Y'	CICS(OPIDENT (xx))	Yes	Yes
	'N'	CICS(NOOPIDENT)	No	Yes
OPCLASS	'Y'	CICS(OPCLASS (xx ...))	Yes	Yes
	'A'	CICS(ADDOPCLASS (xx ...))	No	Yes
	'D'	CICS(DELOPCLASS (xx ...))	No	Yes
	'N'	CICS(NOOPCLASS)	No	Yes
OPPRTY	'Y'	CICS(OOPPRTY (xx))	Yes	Yes
	'N'	CICS(NOOPPRTY)	No	Yes
TIMEOUT	'Y'	CICS(TIMEOUT (xx))	Yes	Yes
	'N'	CICS(NOTIMEOUT)	No	Yes

Table 29. CICS Segment Fields (continued)

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
XRFSOFF	'Y'	CICS(XRFSOFF (xx))	Yes	Yes
	'N'	CICS(NOXRFSOFF)	No	Yes

Table 30. NetView Segment Fields

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
CONSNAME	'Y'	NETVIEW(CONSNAME (xx))	Yes	Yes
	'N'	NETVIEW(NOCONSNAME)	No	Yes
CTL	'Y'	NETVIEW(CTL (xx))	Yes	Yes
	'N'	NETVIEW(NOCTL)	No	Yes
DOMAINS	'Y'	NETVIEW(DOMAINS (xx ...))	Yes	Yes
	'A'	NETVIEW(ADDDOMAINS (xx ...))	No	Yes
	'D'	NETVIEW(DELDOMAINS (xx ...))	No	Yes
	'N'	NETVIEW(NODOMAINS (xx ...))	No	Yes
IC	'Y'	NETVIEW(IC (xx))	Yes	Yes
	'N'	NETVIEW(NOIC)	No	Yes
MSGRECV (Boolean)	'Y'	NETVIEW(MSGRECV (YES))	Yes	Yes
	'N'	NETVIEW(MSGRECV (NO))	Yes	Yes
NGMFADMN (Boolean)	'Y'	NETVIEW(NGMFADMN(YES))	Yes	Yes
	'N'	NETVIEW(NGMFADMN (NO))	No	Yes
NGMFVSPN	'Y'	NETVIEW (NGMFVSPN (xx))	Yes	Yes
	'N'	NETVIEW(NONGMFVSPN)	No	Yes
OPCLASS	'Y'	NETVIEW(OPCLASS (xx ...))	Yes	Yes
	'A'	NETVIEW(ADDOPCLASS (xx ...))	No	Yes
	'D'	NETVIEW(DELOPCLASS (xx ...))	No	Yes
	'N'	NETVIEW(NOOPCLASS)	No	Yes

Table 31. DCE Segment Fields

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
AUTOLOG (Boolean)	'Y'	DCE(AUTOLOGIN (YES))	Yes	Yes
	'N'	DCE(AUTOLOGIN(NO))	Yes	Yes
DCENAME	'Y'	DCE(DCENAME(xx))	Yes	Yes
	'N'	DCE(DCENAME)	No	Yes
HOMECCELL	'Y'	DCE(HOMECCELL (xx))	Yes	Yes
	'N'	DCE(NOHOMECCELL)	No	Yes
HOMEUUID	'Y'	DCE(HOMEUUID (xx))	Yes	Yes
	'N'	DCE(NOHOMEUUID)	No	Yes
UUID	'Y'	DCE(UUID(xx))	Yes	Yes
	'N'	DCE(NOUUID)	No	Yes

Table 32. DFP Segment Fields

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
DATAAPPL	'Y'	DFP(DATAAPPL(xx))	Yes	Yes
	'N'	DFP(NODATAAPPL)	No	Yes
DATACLAS	'Y'	DFP(DATACLAS(xx))	Yes	Yes
	'N'	DFP(NODATACLAS)	No	Yes
MGMCLAS	'Y'	DFP(MGMTCLAS(xx))	Yes	Yes
	'N'	DFP(NOMGMTCLAS)	No	Yes
STORCLAS	'Y'	DFP(STORCLAS(XX))	Yes	Yes
	'N'	DFP(NOSTORCLAS)	No	Yes

Table 33. Language Segment Fields

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
Primary	'Y'	LANGUAGE (PRIMARY(xx))	Yes	Yes
	'N'	LANGUAGE(NOPRIMARY)	No	Yes
Second	'Y'	LANGUAGE(SECONDARY(xx))	Yes	Yes
	'N'	LANGUAGE(NOSECONDARY)	No	Yes

Table 34. OPERPARM Segment Fields

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
ALTGRP	'Y'	OPERPARM (ALTGRP(xx))	Yes	Yes
	'N'	OPERPARM(NOALTGRP)	No	Yes

Table 34. OPERPARM Segment Fields (continued)

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
OPERAUTH	'Y'	OPERPARM(AUTH(xx))	Yes	Yes
	'N'	OPERPARM(NOAUTH)	No	Yes
AUTO	'Y'	OPERPARM(AUTO(xx))	Yes	Yes
	'N'	OPERPARM(NOAUTO))	No	Yes
CMDSYS	'Y'	OPERPARM(CMDSYS (xx))	Yes	Yes
	'N'	OPERPARM (NOCMDSYS)	No	Yes
DOM	'Y'	OPERPARM(DOM xx))	Yes	Yes
	'N'	OPERPARM (NODOM)	No	Yes
KEY	'Y'	OPERPARM(KEY(xx))	Yes	Yes
	'N'	OPERPARM(NOKEY)	No	Yes
LEVEL	'Y'	OPERPARM(LEVEL (xx))	Yes	Yes
	'N'	OPERPARM(NOLEVEL)	No	Yes
LOGCMD	'Y'	OPERPARM(LOGCMDRESP(xx))	Yes	Yes
	'N'	OPERPARM(NOLOGCMDRESP)	No	Yes
MFORM	'Y'	OPERPARM(MFORM(xx))	Yes	Yes
	'N'	OPERPARM(NOMFORM)	No	Yes
MIGID	'Y'	OPERPARM(MIGID(xx))	Yes	Yes
	'N'	OPERPARM(NOMIGID)	No	Yes
MONITOR	'Y'	OPERPARM(xx))	Yes	Yes
	'N'	OPERPARM(NOMONITOR)	No	Yes
MSCOPE	'Y'	OPERPARM(MSCOPE(xx ...))	Yes	Yes
	'A'	OPERPARM(ADDMSCOPE(xx ...))	No	Yes
	'D'	OPERPARM(DELMSCOPE(xx ...))	No	Yes
	'N'	OPERPARM(NOMSCOPE)	No	Yes
ROUTCODE	'Y'	OPERPARM(ROUTCODE(xx ...))	Yes	Yes
	'N'	OPERPARM(NOROUTCODE)	No	Yes
STORAGE	'Y'	OPERPARM(STORAGE(xx))	Yes	Yes
	'N'	OPERPARM(NOSTORAGE)	No	Yes
UD	'Y'	OPERPARM(UD (xx))	Yes	Yes
	'N'	OPERPARM(NOUD)	No	Yes

Table 35. OVM Segment Fields

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
FSROOT	'Y'	OVM(FSROOT (xx))	Yes	Yes
	'N'	OVM(NOFSROOT)	No	Yes
VHOME	'Y'	OVM(HOME(xx))	Yes	Yes
	'N'	OVM(NOHOME)	No	Yes

Table 35. OVM Segment Fields (continued)

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
VPROGRAM	'Y'	OVM(PROGRAM(xx))	Yes	Yes
	'N'	OVM(NOPROGRAM)	No	Yes
VUID	'Y'	OVM(UID(xx))	Yes	Yes
	'N'	OVM(NOUID)	No	Yes

Table 36. WORKATTR Segment Fields

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
WAACCNT	'Y'	WORKATTR(WAACCNT (xx))	Yes	Yes
	'N'	WORKATTR(NOWAACCNT)	No	Yes
WAADDR1	'Y'	WORKATTR(WAADDR1 (xx))	Yes	Yes
	'N'	WORKATTR(NOWADDR1)	No	Yes
WAADDR2	'Y'	WORKATTR(WAADDR2 (xx))	Yes	Yes
	'N'	WORKATTR(NOWADDR2)	No	Yes
WAADDR3	'Y'	WORKATTR(WAADDR3 (xx))	Yes	Yes
	'N'	WORKATTR(NOWADDR3)	No	Yes
WAADDR4	'Y'	WORKATTR(WAADDR4 (xx))	Yes	Yes
	'N'	WORKATTR(NOWADDR4)	No	Yes
WABLDG	'Y'	WORKATTR(WABLDG(xx))	Yes	Yes
	'N'	WORKATTR(NOWABLDG)	No	Yes
WADEPT	'Y'	WORKATTR(WADEPT (xx))	Yes	Yes
	'N'	WORKATTR(NOWADEPT)	No	Yes
WANAME	'Y'	WORKATTR(WANAME (xx))	Yes	Yes
	'N'	WORKATTR(NOWANAME(xx))	No	Yes
WAROOM	'Y'	WORKATTR(WAROOM (xx))	Yes	Yes
	'N'	WORKATTR(NOWAROOM)	No	Yes

Table 37. LNOTES Segment Fields

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
SNAME	'Y'	LNOTES(SNAME (xx))	Yes	Yes
	'N'	LNOTES(NOSNAME)	No	Yes

Table 38. NDS Segment Fields

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
UNAME	'Y'	NDS(UNAME (xx))	Yes	Yes
	'N'	NDS(NOUNAME)	No	Yes

Table 39. KERB Segment Fields

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
KERBNAME	'Y'	KERB(KERBNAME(xx))	Yes	Yes
	'N'	KERB(NOKERBNAME)	No	Yes
MAXTKTLFE	'Y'	KERB(MAXTKTLFE(xx))	Yes	Yes
	'N'	KERB(NOMAXTKTLFE)	No	Yes
ENCRYPT	'Y'	KERB(ENCRYPT(xx))	Yes	Yes
	'N'	KERB(NOENCRYPT)	No	Yes

Table 40. PROXY Segment Fields

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
LDAPHOST	'Y'	PROXY(LDAPHOST(xx))	Yes	Yes
	'N'	PROXY(NOLDAPHOST)	No	Yes
BINDDN	'Y'	PROXY(BINDDN(xx))	Yes	Yes
	'N'	PROXY(NOBINDDN)	No	Yes
BINDPW	'Y'	PROXY(BINDPW(xx))	Yes	Yes
	'N'	PROXY(NOBINDPW)	No	Yes

Table 41. EIM Segment Fields

Field Name	Flag Byte Values	ADDUSER/ALTUSER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
LDAPPROF	'Y'	EIM(LDAPPROF(xx))	Yes	Yes
	'N'	EIM(NOLDAPPROF)	No	Yes

Group administration

For the ADMN_ADD_GROUP, ADMN_DEL_GROUP, ADMN_ALT_GROUP, and ADMN_LST_GROUP function codes, the mapping associated with the function specific parameter list is mapped as follows:

Offset	Length	Description
0	1	Length of the Group Name
1	8	Upper case RACF Group Name
9	1	Reserved
10	2	Output offset to the segment or field entry in error in relation to the start of the Parm_list. Only applied to ADMN_ADD_GROUP and ADMN_ALT_GROUP requests when an "invalid request" error is returned to the caller.

R_admin

Offset	Length	Description
12	2	Number of RACF profile segments
14	*	Start of first segment entry

For ADMN_DEL_GROUP, no segment data is expected. The number of segments should be zero. If non-zero, any segment data present is ignored.

Each segment entry is mapped as follows:

Offset	Length	Description
0	8	Profile segment name (left justified, uppercase, and padded with blanks). For ADMN_LST_GROUP, any GROUP profile segment as defined in the RACF database templates is acceptable. For ADMN_ADD_GROUP and ADMN_ALT_GROUP, the following segments are supported BASE: DFP, OMVS, OVM, and TME.
8	1	Flag byte. 'Y' to create (or alter) the segment. 'N' delete (or not create) the segment.
9	2	Number of fields to update within a segment
11	*	First field for segment

For ADMN_LST_GROUP, the flag byte is ignored.

For ADMN_ADD_GROUP, the flag byte indicates whether the segment should be created ('Y') or not ('N'). Because the BASE segment must always be created, the flag byte for BASE is ignored. In addition, the flag byte of 'I' may be specified for any segment name. When the callable service finds the 'I' flag byte, it ignores the segment.

For ADMN_ALT_GROUP, the flag byte indicates whether the segment should be altered ('Y') or should no longer exist ('N'). Because the BASE segment cannot be deleted, the flag byte for BASE is ignored. In addition, the flag byte of 'I' may be specified for any segment name. When the callable service finds the 'I' flag byte, it ignores the segment.

For ADMN_LST_USER, ADMN_ADD_GROUP, ADMN_ALT_GROUP with flag byte 'N', no field data is expected. The number of fields should be zero. If non-zero, any field data present is ignored.

Each field entry is mapped as follows:

Offset	Length	Description
0	8	Field name as defined below. All field names must be left justified, entered in all uppercase, and padded with blanks.
8	1	Field specific flag byte
9	2	Length of field data
11	*	Field data

For ADMN_ADD_GROUP and ADMN_ALT_GROUP, the following rules apply:

- For boolean fields, the flag byte value of 'Y' or 'N' determines the flag setting. No field data is allowed for boolean fields. Coding field data results in an "Invalid Request" error being returned to the caller.
- For text fields, the flag byte 'Y' indicates the field should be created (or altered) with the specified field data. The flag byte 'N' indicates the field should be deleted (or not created). For flag byte 'N', any field data specified is ignored.
- For repeating text fields, a flag byte of 'A' indicates that the specified field data should be added to the existing data (if any). Field data must be specified. If field data is not specified, an "Invalid Request" error is returned to the caller. The flag byte 'D' indicates that the specified data should be deleted (if any).

In addition to the flag bytes specified in the following tables, a flag byte of 'I' may be specified for any segment name. The 'I' flag byte indicates to the callable service to ignore the field. Any field data specified with a flag byte of 'I' is ignored. In addition, the flag byte of 'I' may be specified for any segment name. When the callable service finds the 'I' flag byte, it ignores the segment.

The following tables define the field keywords and their usage. All field names relate directly to ADDGROUP/ALTGROUP keywords. If you have any questions about field usage and data, refer to *z/OS Security Server RACF Command Language Reference*.

Table 42. BASE Segment Fields

Field Name	Flag Byte Values	ADDGROUP/ALTGROUP Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
SUPGROUP	'Y'	SUPGROUP(xx)	Yes	Yes
OWNER	'Y'	OWNER(xx)	Yes	Yes
TERMUACC (Boolean)	'Y'	TERMUACC	Yes	Yes
	'N'	NOTERMUACC	Yes	Yes
DATA	'Y'	DATA(xx)	Yes	Yes
	'N'	NODATA	No	Yes
MODEL	'Y'	MODEL (xx)	Yes	Yes
	'N'	NOMODEL	No	Yes
UNIVERSL	'Y'	UNIVERSAL	Yes	No

Table 43. DFP Segment Fields

Field Name	Flag Byte Values	ADDGROUP/ALTGROUP Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
DATAAPPL	'Y'	DFP(DATAAPPL(xx))	Yes	Yes
	'N'	DFP(NODATAAPPL)	No	Yes
DATACLAS	'Y'	DFP(DATACLAS(xx))	Yes	Yes
	'N'	DFP(NODATACLAS)	No	Yes
MGMTCLAS	'Y'	DFP(MGMTCLAS(xx))	Yes	Yes
	'N'	DFP(NOMGMTCLAS)	No	Yes
STORCLAS	'Y'	DFP(STORCLAS(xx))	Yes	Yes
	'N'	DFP(NOSTORCLAS)	No	Yes

Table 44. OMVS Segment Fields

Field Name	Flag Byte Values	ADDGROUP/ALTGROUP Keyword Reference	Allowed on Add Requests	Allowed on Alt Requests
GID	'Y'	OMVS(GID(xx))	Yes	Yes
	'N'	OMVS(NOGID)	No	Yes
SHARED	'Y'	OMSV(SHARED)	Yes	Yes
AUTOGID	'Y'	OMVS(AUTOGID)	Yes	Yes

Table 45. OVM Segment Fields

Field Name	Flag Byte Values	ADDGROUP/ALTGROUP Keyword Reference	Allowed on Add Requests	Allowed on Alt Requests
GID	'Y'	OVM(GID(xx))	Yes	Yes
	'N'	OVM(NOGID)	No	Yes

Table 46. TME Segment Fields

Field Name	Flag Byte Values	ADDGROUP/ALTGROUP Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
ROLES	'Y'	TME(ROLES(xx ...))	Yes	Yes
	'A'	TME(ADDROLES(xx ...))	No	Yes
	'D'	TME(DELROLES (xx ...))	No	Yes
	'N'	TME(NOROLES)	No	Yes

Group connection administration: For the ADMN_CONNECT and ADMN_REMOVE function codes the mapping associated with the function specific parameter list is mapped as follows:

Offset	Length	Description
0	1	Length of the User ID1
1	8	Upper case RACF User ID
9	1	Reserved

Offset	Length	Description
10	2	Output offset to the field entry in error in relationship to the start of the Parm_list. Only applies to ADMN_CONNECT and ADMN_REMOVE requests when an "Invalid Request" error is returned to the caller.
12	2	Number of RACF segments
14	*	Start of first segment entry

Each segment entry is mapped as follows:

Offset	Length	Description
0	1	Profile segment name (only the BASE segment is accepted for ADMN_CONNECT and ADMN_REMOVE)
1	8	Flag byte. 'Y' to create (or alter) the segment. 'N' delete (or not create) the segment
9	2	Number of fields to update within a segment.
11	*	First field for segment

By convention, use the BASE segment to specify field information for ADMN_CONNECT and ADMN_REMOVE. The flag byte is ignored. Each field entry is mapped as follows:

Offset	Length	Description
0	1	Field name as defined below. All field names must be left justified, entered in all uppercase, and padded with blanks.
1	8	Field specific flag byte..
9	2	Length of field data
11	*	Field data

For ADMN_CONNECT and ADMN_REMOVE, the following rules apply:

- For boolean fields, the flag byte value of 'Y' or 'N' determines the flag setting. No field data is allowed for boolean fields. Coding field data results in an "Invalid Request" error being returned to the caller.
- For text fields, the flag byte 'Y' indicates the field should be created (or altered) with the specified field data. The flag byte 'N' indicates the field should be deleted (or not created). For flag byte 'N', any field data specified is ignored.
- For repeating text fields, a flag byte of 'A' indicates that the specified field data should be added to the existing data (if any). Field data must be specified. If field data is not specified, an "Invalid Request" error is returned to the caller. The flag byte 'D' indicates that the specified data should be deleted (if any).

In addition to the flag bytes specified in the following tables, a flag byte of 'I' may be specified for any segment name. The 'I' flag byte indicates to the callable service to ignore the field. Any field data specified with a flag byte of 'I' is ignored.

The following table defines the field keywords and their usage. All field names relate directly to the CONNECT and REMOVE keywords. If you have any

R_admin

questions about the field usage and data of these commands, see *z/OS Security Server RACF Command Language Reference*.

Table 47. Base Segment Fields

Field Name	Flag Byte Values	CONNECT and REMOVE Keyword Reference	Allowed on CONNECT Requests	Allowed on REMOVE requests
GROUP	'Y'	GROUP(xx)	Yes	Yes
OWNER	'Y'	OWNER (xx)	Yes	Yes
ADSP (Boolean)	'Y'	ADSP	Yes	No
	'N'	NOADSP	Yes	No
AUDITOR (Boolean)	'Y'	AUDITOR	Yes	No
	'N'	NOAUDITOR	Yes	No
GRPACC (Boolean)	'Y'	GRPACC	Yes	No
	'N'	NOGRPACC	Yes	No
OPER (Boolean)	'Y'	OPERATIONS	Yes	No
	'N'	NOOPERATIONS	Yes	No
REVOKE	'Y'	REVOKE (xx)	Yes	No
RESUME	'Y'	RESUME (xx)	Yes	No
SPECIAL (Boolean)	'Y'	SPECIAL	Yes	No
	'N'	NOSPECIAL	Yes	No
UACC	'Y'	UACC (xx)	Yes	No

Resource profile administration: For all of the resource-related function codes (ADMN_ADD_GENRES, ADMN_ALT_GENRES, ADMN_DEL_GENRES, ADMN_LST_GENRES, ADMN_ADD_DS, ADMN_ALT_DS, ADMN_DEL_DS, ADMN_LST_DS, and ADMN_PERMIT), the mapping associated with the function-specific parameter list is mapped as follows:

Offset	Length	Description
0	1	Length of the class name
1	8	Upper case RACF class name
NOTE: The class name is not required for the data set functions.		
9	1	Reserved
10	2	Output offset to the segment or field entry in error in relation to the start of the Parm_list. Only applies to add, alter, and list requests when an "Invalid Request" error is returned to the caller.
12	2	Number of RACF profile segments
14	*	Start of the first segment entry

Each segment entry is mapped as follows:

Offset	Length	Description
0	8	Profile segment name (left justified, uppercase, and padded with blanks). By convention, use the BASE segment to specify field information for ADMN_PERMIT and the delete and list functions (specify the NORACF field if you do not want the BASE segment listed). To list additional segment information, specify additional segment entries, where any general resource or data set profile segment as defined in the RACF database templates is acceptable. For ADMN_ADD_GENRES and ADMN_ALT_GENRES, the following segments are supported: BASE, DLFDATA, KERB, SESSION, SSIGNON, STDATA, SVFMR, and TME. For ADMN_ADD_DS and ADMN_ALT_DS, the following segments are supported: BASE, DFP, and TME.
8	1	Flag byte. 'Y' to create (or alter) the segment. 'N' delete (or not create) the segment.
9	2	Number of fields to update within a segment
11	*	First field for segment

For ADMN_PERMIT, ADMN_DEL_DS, and for the list functions, the flag byte is ignored.

For the add functions, the flag byte indicates whether the segment should be create ('Y') or not ('N'). Because the BASE segment must always be created, the flag byte for BASE is ignored. In addition, the flag byte of 'I' may be specified for any segment name. When the callable service finds the 'I' flag byte, it ignores the segment.

For the alter functions, the flag byte indicates whether the segment should be altered ('Y') or should no longer exist ('N'). Because the base segment cannot be deleted, the flag byte for BASE is ignored. In addition, the flag byte of 'I' may be specified for any segment name. When the callable service finds the 'I' flag byte, it ignores the segment.

For the add and alter functions with flag byte 'N', no field data is expected. The number of fields should be zero. If non-zero, any field data present is ignored.

Each field entry is mapped as follows:

Offset	Length	Description
0	8	Field name as defined below. All field names must be left justified, entered in all uppercase, and padded with blanks
8	1	Field-specific flag byte
9	2	Length of field data
11	*	Field data

R_admin

For boolean fields, the flag byte value of 'Y' or 'N' determines the flag setting. No field data is allowed for boolean fields. Coding field data results in an "Invalid Request" error being returned to the caller.

For the add and alter functions, the following rules apply:

- For text fields, the flag byte 'Y' indicates the field should be created (or altered) with the specified field data. The flag byte 'N' indicates the field should be deleted (or not created). For flag byte 'N', any field data specified is ignored.
- For repeating text fields, a flag byte of 'A' indicates that the specified field data is to be added to the existing data (if any). Field data must be specified. If field data is not specified an "Invalid Request" error is returned to the caller. The flag byte 'D' indicates that the specified data (if any) is to be deleted.

In addition to the flag bytes specified in the following tables, the flag byte 'I' can be specified for any segment name. The 'I' flag byte indicates to the callable service to ignore the field. Any field data specified with a flag byte of 'I' is ignored.

The general resource, data set, and permit functions each utilize a separate set of field definitions. Table 48 shows where to find the field definitions for the resource related function codes.

Table 48. Resource Related Field Definitions

For Field Definitions	See
ADMN_ADD_GESRES ADMN_ALT_GENRES ADMN_LST_GENRES	"General resource field definitions".
NOTE: For ADMN_DEL_GENRES, specify only the PROFILE field in the BASE segment.	
ADMN_ADD_DS, ADMN_ALT_DS, ADMN_LST_DS, ADMN_DEL_DS	"Data set field definitions" on page 89
ADMN_PERMIT	"Permit field definitions" on page 92

General resource field definitions: The following tables define the general resource field keywords and their usage. All field names relate directly to the RDEFINE, RALTER, and RLIST keywords. If you have questions about the field usage and data, see *z/OS Security Server RACF Command Language Reference*.

Table 49. BASE Segment Fields

Field Name	Flag Byte Value	RDEFINE/RALTER/RLIST Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests	Allowed on List Requests
PROFILE	'Y'	N/A (see note)	Yes	Yes	Yes
NOTE: This field is required. There is no associated command keyword since it is a positional parameter. For ADMN_DEL_GENRES, this is the only allowable field.					
CATEGORY	'Y'	ADDCATEGORY(xx ...)	Yes	No	No
	'A'	ADDCATEGORY(xx ...)	No	Yes	No
	'D'	DELCATEGORY(xx ...)	No	Yes	No
NOTE: To remove unknown categories from the profile, specify the 'D' flag and a field length of zero.					
MEMBER	'Y'	ADDMEM(xx ...)	Yes	No	No
	'N'	ADDMEM(xx ...)	No	Yes	No
	'D'	DELMEM(xx ...)	No	Yes	No

Table 49. BASE Segment Fields (continued)

Field Name	Flag Byte Value	RDEFINE/RALTER/RLIST Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests	Allowed on List Requests
VOLUME	'A'	ADDVOL(xx ...)	No	Yes	No
	'D'	DELVOL(xx ...)	No	Yes	No
APPLDATA	'Y'	APPLDATA(xx)	Yes	Yes	No
	'N'	NOAPPLDATA	No	Yes	No
DATA	'Y'	DATA(xx)	Yes	Yes	No
	'N'	NODATA	No	Yes	No
AUDALTR	'Y'	AUDIT(xx (ALTER))	Yes	Yes	No
AUDCNTRL	'Y'	AUDIT(xx (CONTROL))	Yes	Yes	No
AUDREAD	'Y'	AUDIT(xx (READ))	Yes	Yes	No
AUDUPDT	'Y'	AUDIT(xx (UPDATE))	Yes	Yes	No
AUDNONE	'Y'	AUDIT(NONE)	Yes	Yes	No
GAUDALTR	'Y'	GLOBALAUDIT(xx (ALTER))	No	Yes	No
GAUDCNTR	'Y'	GLOBALAUDIT(xx (CONTROL))	No	Yes	No
GAUDREAD	'Y'	GLOBALAUDIT(xx (READ))	No	Yes	No
GAUDUPDT	'Y'	GLOBALAUDIT(xx UPDATE))	No	Yes	No
GAUDNONE	'Y'	GLOBALAUDIT (NONE)	No	Yes	No
FPROFILE	'Y'	FROM(xx)	Yes	No	No
FCLASS	'Y'	FCLASS(xx)	Yes	No	No
FVOLUME	'Y'	FVOLUME(xx)	Yes	No	No
FGENERIC (Boolean)	'Y'	FGENERIC	Yes	No	No
LEVEL	'Y'	LEVEL(xx)	Yes	Yes	No
NOTIFY	'Y'	NOTIFY(xx)	Yes	Yes	No
	'N'	NONOTIFY	No	Yes	No
OWNER	'Y'	OWNER(xx)	Yes	Yes	No
SECLABEL	'Y'	SECLABEL (xx)	Yes	Yes	No
	'N'	NOSECLABEL	No	Yes	No
SECLEVEL	'Y'	SECLEVEL(xx)	Yes	Yes	No
	'N'	NOSECLEVEL	No	Yes	No
SINGLDSN (Boolean)	'Y'	SINGLDSN	Yes	Yes	No
	'N'	NOSINGLDSN	No	Yes	No
TIMEZONE	'Y'	TIMEZONE(xx)	Yes	Yes	No
	'N'	NOTIMEZONE	No	Yes	No
TVTOC (Boolean)	'Y'	TVTOC	Yes	Yes	Yes
	'N'	NOTVTOC	No	Yes	No
UACC	'Y'	UACC(xx)	Yes	Yes	No

Table 49. BASE Segment Fields (continued)

Field Name	Flag Byte Value	RDEFINE/RALTER/RLIST Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests	Allowed on List Requests
WARNING (Boolean)	'Y'	WARNING	Yes	Yes	No
	'N'	NOWARNING	No	Yes	No
WHENDAYS	'Y'	WHEN(DAYS(xx))	Yes	Yes	No
WHENTIME	'Y'	WHEN(TIME (xx ...))	Yes	Yes	No
ALL (Boolean)	'Y'	ALL	No	No	Yes
AUTHUSER (Boolean)	'Y'	AUTHUSER	No	No	Yes
GENERIC	'Y'	GENERIC	No	No	Yes
	'N'	NOGENERIC	No	No	Yes
HISTORY (Boolean)	'Y'	HISTORY	No	No	Yes
NORACF (Boolean)	'Y'	NORACF	No	No	Yes
NOYOURAC (Boolean)	'Y'	NOYOURACC	No	No	Yes
RESGROUP	'Y'	RESGROUP	No	No	Yes
STATS (Boolean)	'Y'	STATISTICS	No	No	Yes

Table 50. DLFDATA Segment Fields

Field Name	Flag Byte Values	RDEFINE/RALTER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
RETAIN (Boolean)	'Y'	DLFDATA(RETAIN(YES))	Yes	Yes
	'N'	DLFDATA (RETAIN(NO))	Yes	Yes
JOBNAME	'Y'	DLFDATA (JOBNAMES (...))	Yes	Yes
	'A'	DLFDATA(ADDJOBNAMES(...))	No	Yes
	'D'	DLFDATA(DELJOBNAMES(...))	No	Yes
	'N'	DLFDATA(NOJOBNAMES)	No	Yes

Table 51. SESSION Segment Fields

Field Name	Flag Byte Value	RDEFINE/RALTER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
CONVSEC	'Y'	SESSION (CONVSEC(...))	Yes	Yes
	'N'	SESSION (NOCONVSEC))	No	Yes
INTERVAL	'Y'	SESSION(INTERVAL(...))	Yes	Yes
	'N'	SESSION (NOINTERVAL))	No	Yes
LOCK	'Y'	SESSION (LOCK)	Yes	Yes
	'N'	SESSION (NOLOCK)	No	Yes

Table 51. SESSION Segment Fields (continued)

Field Name	Flag Byte Value	RDEFINE/RALTER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
SESSKEY	'Y'	SESSION (SESSKEY(...))	Yes	Yes
	'N'	SESSION(NOSESSKEY))	No	Yes

Table 52. SSIGNON Segment Fields

Field Name	Flag Byte Values	REDEFINE/RALTER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
KEYMASK	'Y'	SSIGNON (KEYMASK(...))	Yes	Yes
	'N'	SSIGNON (NOKEYMASK)	No	Yes
KEYCRYPT	'Y'	SSIGNON (KEYENCRYPT (...))	Yes	Yes
	'N'	SSIGNON(NOKEYENCRYPT)	No	Yes

Table 53. STDATA Segment Fields

Field Name	Flag Byte Values	RDEFINE/RALTER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
USER	'Y'	STDATA(USER(...))	Yes	Yes
	'N'	STDATA(NOUSER)	No	Yes
GROUP	'Y'	STDATA(GROUP(...))	Yes	Yes
	'N'	STDATA(NOGROUP)	No	Yes
PRIVILEGE (Boolean)	'Y'	STDATA(PRIVILEGED(YES))	Yes	Yes
	'N'	STDATA(PRIVILEGED(NO))	Yes	Yes
TRACE (Boolean)	'Y'	STDATA(TRACE(YES))	Yes	Yes
	'N'	STDATA(TRACE(NO))	Yes	Yes
TRUSTED (Boolean)	'Y'	STDATA(TRUSTED(YES))	Yes	Yes
	'N'	STDATA(TRUSTED(NO))	Yes	Yes

Table 54. SVFMR Segment Fields

Field Name	Flag Byte Values	RDEFINE/RALTER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
SCRIPTNAME	'Y'	SVFMR(SCRIPTNAME(...))	Yes	Yes
	'N'	SVFMR(NOSCRIPTNAME)	No	Yes
PARMNAME	'Y'	SVFMR(PARMNAME(...))	Yes	Yes
	'N'	SVFMR(NOPARMNAME)	No	Yes

Table 55. TME Segment Fields

Field Name	Flag Byte Values	RDEFINE/RALTER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
ROLES	'Y'	TME(ROLES(xx ...))	Yes	Yes
	'A'	TME(ADDROLES(xx ...))	No	Yes
	'D'	TME(DELROLES(xx ...))	No	Yes
	'N'	TME(NOROLES)	No	Yes
GROUPS	'Y'	TME(GROUPS(xx ...))	Yes	Yes
	'A'	TME(ADDGROUPS(xx ...))	No	Yes
	'D'	TME(DELGROUPS(xx ...))	No	Yes
	'N'	TME(NOGROUPS)	No	Yes
RESOURCE	'Y'	TME(RESOURCE(xx ...))	Yes	Yes
	'A'	TME(ADDRESOURCE(xx ...))	No	Yes
	'D'	TME(DELRESOURCE(xx ...))	No	Yes
	'N'	TME(NORESOURCE)	No	Yes
CHILDREN	'Y'	TME(CHILDREN(xx ...))	Yes	Yes
	'A'	TME(ADDCHILDREN(xx ...))	No	Yes
	'D'	TME(DELCHILDREN(xx ...))	No	Yes
	'N'	TME(NOCHILDREN)	No	Yes
PARENT	'Y'	TME(PARENT(xx ...))	Yes	Yes
	'N'	TME(NOPARENT)	No	Yes

Table 56. KERB Segment Fields

Field Name	Flag Byte Values	RDEFINE/RALTER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
KERBNAME	'Y'	KERB(KERBNAME(xx))	Yes	Yes
	'N'	KERB(NOKERBNAME)	No	Yes
DEFTKTLF	'Y'	KERB(DEFTKTLFE(xx))	Yes	Yes
	'N'	KERB(NODEFTKTLFE)	No	Yes
MAXTKTLFE	'Y'	KERB(MAXTKTLFE(xx))	Yes	Yes
	'N'	KERB(NOMAXTKTLFE)	No	Yes
MINTKTLF	'Y'	KERB(MINTKTLFE(xx))	Yes	Yes
	'N'	KERB(NOMINTKTLFE)	No	Yes
PASSWORD	'Y'	KERB(PASSWORD(xx))	Yes	Yes
	'N'	KERB(NOPASSWORD)	No	Yes
ENCRYPT	'Y'	KERB(ENCRYPT(xx))	Yes	Yes
	'N'	KERB(NOENCRYPT)	No	Yes

Table 57. PROXY Segment Fields

Field Name	Flag Byte Values	RDEFINE/RALTER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
LDAPHOST	'Y'	PROXY(LDAPHOST(xx))	Yes	Yes
	'N'	PROXY(NOLDAPHOST)	No	Yes
BINDDN	'Y'	PROXY(BINDDN(xx))	Yes	Yes
	'N'	PROXY(NOBINDDN)	No	Yes
BINDPW	'Y'	PROXY(BINDPW(xx))	Yes	Yes
	'N'	PROXY(NOBINDPW)	No	Yes

Table 58. EIM Segment Fields

Field Name	Flag Byte Values	RDEFINE/RALTER Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
DOMAINDN	'Y'	EIM(DOMAINDN(xx))	Yes	Yes
	'N'	EIM(NODOMAINDN)	No	Yes
OPTIONS	'Y'	EIM(OPTIONS(xx))	Yes	Yes
	'N'	EIM(NOOPTIONS)	No	Yes
LOCALREG	'Y'	EIM(LOCALREGISTRY(xx))	Yes	Yes
	'N'	EIM(NOLOCALREGISTRY)	No	Yes

Data set field definitions: The following table defines the DATASET field keywords and their usage. All field names relate directly to ADDSD, ALTDSD, DELDSD, and LISTSD keywords. If you have any questions about field usage and data, see *z/OS Security Server RACF Command Language Reference*.

Table 59. BASE Segment Fields

Field Name	Flag Byte Value	ADDS, ALTDSD, DELDSD, LISTSD Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests	Allowed on List Requests	Allowed on Delete Requests
PROFILE	'Y'	for list, DATASET (xx...)	Yes	Yes	Yes	Yes
NOTE: For add, alter, and delete, this field is required. This is no associated command keyword because it is a positional parameter. For list, this field is optional and is used in the DATASET(xx...) keyword.						
For add and alter, if working with a password-protected data set, append "/password?" to the data set profile name, and include this additional data in the length field for the data set profile name.						
CATEGORY	'Y'	ADDCATEGORY (xx ...)	Yes	No	No	No
	'A'	ADDCATEGORY (xx ...)	No	Yes	No	No
	'D'	DELCATEGORY (xx ...)	No	Yes	No	No

Table 59. BASE Segment Fields (continued)

Field Name	Flag Byte Value	ADDSD, ALTSD, DELSD, LISTSD Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests	Allowed on List Requests	Allowed on Delete Requests
NOTE: To remove unknown categories from the profile, specify the 'D' flag and a field length of zero.						
VOLUME	'Y'	VOLUME (xx ...)	Yes	Yes	Yes	Yes
	'A'	ADDVOL (xx ...)	No	Yes	No	
	'D'	DELVOL (xx ...)	No	Yes	No	
ALTVOL	'Y'	ALTVOL (xx)	No	Yes	No	No
GENERIC (Boolean)	'Y'	GENERIC	Yes	Yes	Yes	Yes
	'N'	NOGENERIC	No	No	Yes	No
MODEL (Boolean)	'Y'	MODEL	Yes	No	No	No
TAPE (Boolean)	'Y'	TAPE	Yes	No	No	No
SET (Boolean)	'Y'	SET	Yes	Yes	No	Yes
SETONLY (Boolean)	'Y'	SETONLY	Yes	No	No	No
AUDALTR	'Y'	AUDIT(xx(ALTER))	Yes	Yes	No	No
AUDCNTR	'Y'	AUDIT (xx (CONTROL))	Yes	Yes	No	No
AUDREAD	'Y'	AUDIT (xx (READ))	Yes	Yes	No	No
AUDUPDT	'Y'	AUDIT (xx (UPDATE))	Yes	Yes	No	No
AUDNONE (Boolean)	'Y'	AUDIT (NONE)	Yes	Yes	No	No
GAUDALTR	'Y'	GLOBALAUDIT (xx (ALTER))	No	Yes	No	No
GAUDCNTR	'Y'	GLOBALAUDIT (xx (CONTROL))	No	Yes	No	No
GAUDREAD	'Y'	GLOBALAUDIT (xx (READ))	No	Yes	No	No
GAUDUPDT	'Y'	GLOBALAUDIT (xx UPDATE))	No	Yes	No	No
GAUDNONE (Boolean)	'Y'	GLOBALAUDIT (NONE)	No	Yes	No	No
DATA	'Y'	DATA (xx)	Yes	Yes	No	No
	'N'	NODATA	No	Yes	No	No
ERASE (Boolean)	'Y'	ERASE	Yes	Yes	No	No
	'N'	NOERASE	No	Yes	No	No
FILESEQ	'Y'	FILESEQ (xx)	Yes	No	No	No
FROM	'Y'	FROM (xx)	Yes	No	No	No
FCLASS	'Y'	FCLASS (xx)	Yes	No	No	No

Table 59. BASE Segment Fields (continued)

Field Name	Flag Byte Value	ADDSD, ALTDSD, DELDSD, LISTDSD Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests	Allowed on List Requests	Allowed on Delete Requests
FVOLUME	'Y'	FVOLUME (xx)	Yes	No	No	No
FGENERIC (Boolean)	'Y'	FGENERIC	Yes	No	No	No
LEVEL	'Y'	LEVEL (xx)	Yes	Yes	No	No
NOTIFY	'Y'	NOTIFY (xx)	Yes	Yes	No	No
	'N'	NONOTIFY	No	Yes	No	No
OWNER	'Y'	OWNER (xx)	Yes	Yes	No	No
SECLABLE	'Y'	SECLABEL (xx)	Yes	Yes	No	No
	'N'	NOSECLABEL	No	Yes	No	No
SECLEVEL	'Y'	SECLEVEL (xx)	Yes	Yes	No	No
	'N'	NOSECLEVEL	No	Yes	No	No
UACC	'Y'	UACC	Yes	Yes	No	No
UNIT	'Y'	UNIT (xx)	Yes	Yes	No	No
WARNING (Boolean)	'Y'	WARNING	Yes	Yes	No	No
	'N'	NOWARNING	No	Yes	No	No
PREFIX	'Y'	PREFIX (xx)	No	No	Yes	No
ALL (Boolean)	'Y'	ALL	No	No	Yes	No
AUTHUSER (Boolean)	'Y'	AUTHUSER	No	No	Yes	No
DSNS (Boolean)	'Y'	DSNS	No	No	Yes	No
HISTORY	'Y'	HISTORY	No	No	Yes	No
NORACF (Boolean)	'Y'	NORACF	No	No	Yes	No
STATS (Boolean)	'Y'	STATISTICS	No	No	Yes	No
RETPD	'Y'	RETPD (xx)	Yes	Yes	No	No

Table 60. DFP Segment Fields

Field Name	Flag Byte Value	ADDSD and ALTDSD Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
RESOWNER	'Y'	DFP(RESOWNER(xx))	Yes	Yes
	'N'	DFP(NORESOWNER)	No	Yes

Table 61. TME Segment Fields

Field Name	Flag Byte Value	ADDSD and ALTDSD Keyword Reference	Allowed on Add Requests	Allowed on Alter Requests
ROLES	'Y'	TME(ROLES(xx...))	Yes	Yes
	'A'	TME(ADDRoles(xx...))	No	Yes
	'D'	TME(DELROLES(xx...))	No	Yes
	'N'	TME(NORoles)	No	Yes

Permit field definitions: The following table defines the permit field keywords and their usage. All field names relate directly to PERMIT keywords. For information on field usage and data, see *z/OS Security Server RACF Command Language Reference*.

Table 62. Base Segment Fields

Field Name	Flag Byte Values	PERMIT Keyword Reference
PROFILE	'Y'	N/A (See note.)
NOTE: This field is required; there is no associated command keyword since it is a positional parameter.		
ACCESS	'Y'	ACCESS (xx)
DELETE (Boolean)	'Y'	DELETE
FPROFILE	'Y'	FROM (xx)
FCLASS	'Y'	FCLASS (xx)
FVOLUME	'Y'	FVOLUME (xx)
FGENERIC (Boolean)	'Y'	FGENERIC
GENERIC (Boolean)	'Y'	GENERIC
ID	'Y'	ID (xx)
RESET	'Y'	RESET (xx)
VOLUME	'Y'	VOLUME (xx)
WHENAPPC	'Y'	WHEN (APPCPORT (...))
WHENCONS	'Y'	WHEN (CONSOLE (...))
WHENJES	'Y'	WHEN (JESINPUT (...))
WHENPROG	'Y'	WHEN (PROGRAM (...))
WHENTERM	'Y'	WHEN (TERMINAL (...))
WHENSYs	'Y'	WHEN (SYSID (...))

Parameter list format for SETROPTS administration: For the ADMN_ALT_SETR function code, the function-specific parameter list is mapped as follows:

Table 63. Parameter List Mapping for SETROPTS Administration

Offset	Length	Description
0	10	Reserved

Table 63. Parameter List Mapping for SETROPTS Administration (continued)

Offset	Length	Description
10	2	Output offset to the segment or field entry in error, relative to the start of the Parm_list. Only applies when an "Invalid Request" error is returned to the caller.
12	2	Number of RACF profile segments
14	*	Start of first segment entry

The segment entry is mapped as follows:

Table 64. Segment Entry Fields

Offset	Length	Description
0	8	Profile segment name (only the "BASE" segment is accepted for ADMN_ALT_SETR)
8	1	Flag byte. Ignored.
9	2	Number of fields to update within a segment
11	*	First field for segment

Each field entry is mapped as follows:

Table 65. Field Entry Format

Offset	Length	Description
0	8	Field name as defined below. All field names must be left justified, entered in all uppercase, and padded with blanks
8	1	Field-specific flag byte
9	2	Length of field data
11	*	Field data

The following rules apply:

- For boolean fields, the flag byte value of 'Y' or 'N' determines the flag setting. No field data is allowed for boolean fields. Coding field data results in an "Invalid Request" error being returned to the caller.
- For text fields, the flag byte 'Y' indicates the field should be created (or altered) with the specified field data. The flag byte 'N' indicated the field should be deleted (or not created). For flag byte 'N', any field data specified is ignored.
- For repeating text fields, a flag byte of 'A' indicates to add the specified field data to the existing data (if any). Field data must be specified, otherwise an "Invalid Request" error is returned to the caller. The flag byte 'D' indicates to delete the specified data (if any).

In addition to the flag bytes specified in the following tables, a flag byte of 'I' can be specified for any segment name. The callable service ignores any field data specified with a flag byte of 'I'.

The following table defines the SETROPTS field keywords and their usage. All field names relate directly to SETROPTS keywords. Refer to z/OS Security Server (RACF) Command Language Reference for information about SETROPTS

keywords. Note that within the command image generated internally, RACF truncates long keywords to 12 characters.

Table 66. BASE Segment Field Names

Field Name	Flag Byte Value	SETROPTS Keyword Reference
CLASSACT	'A'	CLASSACT (xx ...)
	'D'	NOCLASSACT (xx ...)
CLASSTAT	'A'	STATISTICS (xx ...)
	'D'	NOSTATISTICS (xx ...)
EIMREG	'Y'	EIMREGISTRY
	'N'	NOEIMREGISTRY
GENCMD	'A'	GENCMD (xx ...)
	'D'	NOGENCMD (xx ...)
GENERIC	'A'	GENERIC (xx ...)
	'D'	NOGENERIC (xx ...)
GENLIST	'A'	GENLIST (xx ...)
	'D'	NOGENLIST (xx ...)
GLOBAL	'A'	GLOBAL (xx ...)
	'D'	NOGLOBAL (xx ...)
RACLIST	'A'	RACLIST (xx ...)
	'D'	NORACLIST (xx ...)
INACTIVE	'Y'	INACTIVE (xx ...)
	'N'	NOINACTIVE (xx ...)
INITSTAT	'Y'	INITSTATS (xx ...)
	'N'	NOINITSTATS (xx ...)
TERMINAL	'Y'	TERMINAL(xx)
AUDIT	'A'	AUDIT (xx ...)
	'D'	NOAUDIT (xx ...)
CMDVIOL (Boolean)	'Y'	CMDVIOL
	'N'	NOCMDVIOL
OPERAUDT (Boolean)	'Y'	OPERAUDT
	'N'	NOOPERAUDT
SAUDIT (Boolean)	'Y'	SAUDIT
	'N'	NOSAUDIT
APPLAUDT (Boolean)	'Y'	APPLAUDIT
	'N'	NOAPPLAUDIT
SLABAUDT (Boolean)	'Y'	SECLABELAUDIT
	'N'	NOSECLABELAUDIT
SLEVAUDT	'Y'	SECLEVELAUDIT (xx ...)
	'N'	NOSECLEVELAUDIT
KERBLVL	'Y'	KERBLVL(x)
LOGALWYS	'Y'	LOGOPTIONS (ALWAYS (xx ...))

Table 66. BASE Segment Field Names (continued)

Field Name	Flag Byte Value	SETROPTS Keyword Reference
LOGNEVER	'Y'	LOGOPTIONS (NEVER (xx ...))
LOGSUCC	'Y'	LOGOPTIONS (SUCCESSSES (xx ...))
LOGFAIL	'Y'	LOGOPTIONS (FAILURES (xx ...))
LOGDEFLT	'Y'	LOGOPTIONS (DEFAULT (xx ...))
HISTORY	'Y'	PASSWORD (HISTORY (xx))
	'N'	PASSWORD (NOHISTORY)
INTERVAL	'Y'	PASSWORD (INTERVAL (xx))
REVOKE	'Y'	PASSWORD (REVOKE (xx))
	'N'	PASSWORD (NOREVOKE)
WARNING	'Y'	PASSWORD (WARNING (xx))
	'N'	PASSWORD (NOWARNING)
RULES (Boolean)	'N'	PASSWORD (NORULES)
NOTE: Specifying RULES with the 'N' flag results in the cancellation of all password syntax rules, regardless of any RULEn fields also specified.		
RULE1	'Y'	PASSWORD (RULE1 (LENGTH (m1:m2) content-keyword (position)))
	'N'	PASSWORD (NORULE1)
RULE2	'Y'	PASSWORD (RULE2 (LENGTH (m1:m2) content-keyword (position)))
	'N'	PASSWORD (NORULE2)
RULE3	'Y'	PASSWORD (RULE3 (LENGTH (m1:m2) content-keyword (position)))
	'N'	PASSWORD (NORULE3)
RULE4	'Y'	PASSWORD (RULE4 (LENGTH (m1:m2) content-keyword (position)))
	'N'	PASSWORD (NORULE4)
RULE5	'Y'	PASSWORD (RULE5 (LENGTH (m1:m2) content-keyword (position)))
	'N'	PASSWORD (NORULE5)
RULE6	'Y'	PASSWORD (RULE6 (LENGTH (m1:m2) content-keyword (position)))
	'N'	PASSWORD (NORULE6)
RULE7	'Y'	PASSWORD (RULE7 (LENGTH (m1:m2) content-keyword (position)))
	'N'	PASSWORD (NORULE7)
RULE8	'Y'	PASSWORD (RULE8 (LENGTH (m1:m2) content-keyword (position)))
	'N'	PASSWORD (NORULE8)

Table 66. BASE Segment Field Names (continued)

Field Name	Flag Byte Value	SETROPTS Keyword Reference
<p>NOTE: When specifying the 'Y' flag, the data supplied in the RULEn field consists of a length field and a character sequence, separated by a blank. The length field can be either a single numeric value, or two numeric values separated by a colon (:) to denote a minimum and maximum length. The character sequence conforms to the format of the output of the SETROPTS LIST command. It is a string of 1 to 8 characters, where each position of the string contains a character that indicates the valid characters that can occupy that position:</p> <ul style="list-style-type: none"> • A - Alphabetic • C - Consonant • L - Alphanumeric • N - Numeric • V - Vowel • W - Non-vowel • * - Any character <p>For example, if the RULE1 field is specified with field data of "3:6 A*NV*A", the resulting SETROPTS PASSWORD keyword would be RULE1(LENGTH(3:6) ALPHA(1 6) NUMERIC(3) VOWEL(4)).</p> <p>See the <i>z/OS Security Server RACF Command Language Reference</i> for details on SETROPTS.</p>		
LIST (Boolean)	'Y'	LIST
NOTE: The LIST field is not returned by ADMN_UNL_SETR or ADMN_XTR_SETR.		
ADDCREAT (Boolean)	'Y'	ADDCREATOR
	'N'	NOADDCREATOR
ADSP (Boolean)	'Y'	ADSP
	'N'	NOADSP
CATDSNS	'Y'	CATDSNS (xx)
	'N'	NOCATDSNS
COMPMODE (Boolean)	'Y'	COMPATMODE
	'N'	NOCOMPATMODE
EGN (Boolean)	'Y'	EGN
	'N'	NOEGN
GENOWNER (Boolean)	'Y'	GENERICOWNER
	'N'	NOGENERICOWNER
GRPLIST (Boolean)	'Y'	GRPLIST
	'N'	NOGRPLIST
MLACTIVE	'Y'	MLACTIVE (xx)
	'N'	NOMLACTIVE
MLQUIET (Boolean)	'Y'	MLQUIET
	'N'	NOMLQUIET
MLS	'Y'	MLS (xx)
	'N'	NOMLS
MLSTABLE (Boolean)	'Y'	MLSTABLE
	'N'	NOMLSTABLE

Table 66. BASE Segment Field Names (continued)

Field Name	Flag Byte Value	SETROPTS Keyword Reference
PREFIX	'Y'	PREFIX (xx)
	'N'	NOPREFIX
PROTALL	'Y'	PROTECTALL (xx)
	'N'	NOPROTECTALL
REALDSN (Boolean)	'Y'	REALDSN
	'N'	NOREALDSN
REFRESH (Boolean)	'Y'	REFRESH
NOTE: The REFRESH field is not returned by ADMN_UNL_SETR or ADMN_XTR_SETR.		
RETPD	'Y'	RETPD (xx)
RVARSWPW	'Y'	RVARY (SWITCH (xx))
NOTE: For ADMN_XTR_SETR, the value returned for this field is not the actual password, but one of two predefined values. A value of "DEFAULT" indicates that the default password is in effect, while a value of "INSTLN" indicates that an installation-defined password is in effect.		
RVARSTPW	'Y'	RVARY (STATUS (xx))
NOTE: For ADMN_XTR_SETR, the value returned for this field is not the actual password, but one of two predefined values. A value of "DEFAULT" indicates that the default password is in effect, while a value of "INSTLN" indicates that an installation-defined password is in effect.		
SECLABCT (Boolean)	'Y'	SECLABELCONTROL
	'N'	NOSECLABELCONTROL
SESSINT	'Y'	SESSIONINTERVAL (xx)
	'N'	NOSESSIONINTERVAL
TAPEDSN (Boolean)	'Y'	TAPEDSN
	'N'	NOTAPEDSN
WHENPROG (Boolean)	'Y'	WHEN (PROGRAM)
	'N'	NOWHEN (PROGRAM)
MODGDG (Boolean)	'Y'	MODEL (GDG)
	'N'	MODEL (NOGDG)
MODGROUP (Boolean)	'Y'	MODEL (GROUP)
	'N'	MODEL (NOGROUP)
MODUSER (Boolean)	'Y'	MODEL (USER)
	'N'	MODEL (NOUSER)
MODEL (Boolean)	'N'	NOMODEL
ERASE (Boolean)	'Y'	ERASE
	'N'	NOERASE
ERASEALL (Boolean)	'Y'	ERASE (ALL)
ERASESEC	'Y'	ERASE (SECLEVEL (xx))
	'N'	ERASE (NOSECLEVEL)

Table 66. BASE Segment Field Names (continued)

Field Name	Flag Byte Value	SETROPTS Keyword Reference
PRIMLANG	'Y'	LANGUAGE (PRIMARY (xx))
SECLANG	'Y'	LANGUAGE (SECONDARY (xx))
JESBATCH (Boolean)	'Y'	JES (BATCHALLRACF)
	'N'	JES (NOBATCHALLRACF)
JESEARLY (Boolean)	'Y'	JES (EARLYVERIFY)
	'N'	JES (NOEARLYVERIFY)
JESXBM (Boolean)	'Y'	JES (XBMALLRACF)
	'N'	JES (NOXBMALLRACF)
JESNJE	'Y'	JES (NJEUSERID(xx))
JESUNDEF	'Y'	JES (UNDEFINEDUSER(xx))

Output message block: Following is the mapping of the output message block returned by R_admin for the ADMN_XTR_SETR and ADMN_UNL_SETR function codes. The output storage is obtained in the subpool specified by the caller in the Out_message_subpool parameter of IRRSEQ00.

Table 67. Output Message Block

Offset	Length	Description
0	4	Eye catcher to aid in virtual storage dumps: 'RXTR' or 'RUNL'
4	4	Total length of output buffer
8	4	Reserved
12	2	Number of segment entries for ADMN_XTR_SETR, or number of record types returned for ADMN_UNL_SETR.
12	2	Number of RACF profile segments
14	0	Start of the first segment or record entry

For ADMN_XTR_SETR, the output consists of a single segment entry for the base segment, followed by field entries for each of the supported input fields documented in "Parameter list format for SETROPTS administration" on page 92. Note that not all of the fields are returned, and fields that are not returned are noted in the field table. There is no defined order in which fields are returned. The segment and field entry for ADMN_XTR_SETR uses the standard ADMN_USRADM_SEGENTRY and ADMN_USRADM_FLDENTRY mappings used by other R_admin functions.

For ADMN_UNL_SETR, the output data is mapped using the following mapping for each unloaded record type, the number of which is contained in ADMN_EXTRACT_NUM. There is a single record type of "RACFINIT" to describe the basic RACF options. Following this record is a series of records of type "CLASNAME". There are as many "CLASNAME" records as there are classes defined in the class descriptor table (CDT). These include installation-defined classes as well as classes supplied by IBM. Columns 44 through 51 of each record identify the name of the class that the record describes. See *z/OS Security Server RACF Macros and Interfaces* for detailed mappings of these record types. Note that

for ADMN_UNL_SETR, R_admin does not fill in the time-written, date-written, and SMF system ID fields.

Offset	Length	Description
0	8	The SMF data unload record type (documented in <i>z/OS Security Server RACF Macros and Interfaces</i>).
8	4	The length of an individual record of this type.
12	4	The number of records of this type.
16	8	Reserved
24	*	The start of the first record of this type.

R_audit (IRRSAU00): Provide an audit interface

Function

The **R_audit** service provides an audit interface for functions that need to write an audit record for a condition where an audit by a security check service is not sufficient.

This service fills in the base part of the record and some standard relocate sections based on the function code in the CRED and the defined input parameters.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF authorization

None

Format

```
CALL IRRSAU00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, CRED,
               ALET, File_Identifier_1,
               ALET, FSP1,
               ALET, File_deleted_flag,
               ALET, File_Identifier_2,
               ALET, FSP2
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

CRED

The name of the CRED structure for the current file system syscall.

File_Identifier_1

The name of a 16-byte area containing a unique identifier of the file identified by the old (or only) pathname specified on the syscall.

FSP1

The name of the IFSP for the old (or only) file.

File_deleted_flag

For system calls that can cause a file to be deleted, the address of a byte containing a flag:

- 0 - the last link was not removed.
- 1 - the last link was removed for a file. The file is deleted.

File_Identifier_2

For system calls that create a new file name. If the "new" file existed, this is the name of a 16-byte area containing a unique identifier of the "new" file.

FSP2

The name of the IFSP for the new file.

Return and reason codes

IRRSAU00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	24	The audit function code is not valid.
8	8	32	The CRED user type is not supported.

Usage notes

1. This service can be called by the MVS BCP, by a z/OS UNIX file system, or by z/OS UNIX servers. The service contains support for z/OS UNIX servers, but can not be directly invoked by an z/OS UNIX server.
2. IRRSAU00 tests whether auditing is required and, if so, builds and writes an audit record. The record built contains data from the calling process's security attributes (USP) and from the input CRED, the input IFSPs, and the input parameters. The content depends on the function being audited, as determined from the CRED_audit_function_code.
3. See *z/OS Security Server RACF Macros and Interfaces* for tables describing the data included in audit records, the data included in each event record, and syscalls that cause the event records to be written.

Related services

None

R_cacheserv (IRRSCH00): Cache Services

Function

The **R_cacheserv** SAF callable service provides a mechanism for the storage and retrieval of security relevant information from a cache. In addition, the entire cache can be automatically hardened to the security product (RACF) database, and restored as needed.

Requirements

Authorization:	Any PSW key in supervisor state or problem state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have a FRR active
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. The words containing the ALETs must be in the primary address space.

R_cacheserv

Linkage conventions

The parameter list for this callable service is intended to be variable length to allow for future expansion. Use the **NumParms** parameter to indicate the number of parameters specified.

RACF authorization

For callers not running in system key or supervisor state, the use of R_cacheserv is authorized by the resource IRR.RCACHESERV.*cachename* in the FACILITY class. The application server must be running with a RACF user or group ID that has at least READ authority to this resource. READ allows the application server to utilize the **Fetch** function, x'0004', while UPDATE authority provides the capability to use all the functions. If the FACILITY class is inactive or the resource is not defined, only servers running with a system key or in supervisor state may use the R_cacheserv service. Changes to the RACF user or group ID's authorization to the facility class profile, IRR.RCACHESERV.*cachename*, will not take effect until the job step task invoking the R_cacheserv service ends and a new one is started.

Format

```
CALL IRRSCH00 (Work_area,  
              ALET, SAF_return_code,  
              ALET, RACF_return_code,  
              ALET, RACF_reason_code,  
              ParmALET,  
              NumParms,  
              Function_code,  
              Option,  
              Version,  
              Version_length,  
              Cache_name,  
              Record_name_ptr,  
              Record_name_length,  
              Data_ptr,  
              Data_length  
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF. The work area must be in the primary address space.

ALET

The name of a fullword containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

ParmALET

The name of a fullword which must be in the primary address space and contains the ALET for the remaining parameters.

NumParms

The name of a fullword containing the number of remaining parameters in the parameter list, including the NumParms parameter. This number must be 10 for z/OS Version 1, Release 3.

Function_code

The name of a half word (2 byte) area containing the function code. The function code has one of the following values:

X'0001'-Start a new cache.

The cache is created and the caller is now ready to start adding records. No one but the caller has access to the cache. The cache is not made available to other callers at this time.

X'0002'-Add a record to the new cache.

The caller must provide a name for this record and the data associated with the record. The caller calls this function multiple times, once per record to complete the cache. Only the caller of function code X'0001' may call function code X'0002'. A caller of function code X'0002' who is not the same task as the one who called function code X'0001', will not be allowed to add a record to the cache.

X'0003'-End cache creation.

Only the caller of function code X'0001' is allowed to end the creation of the cache. A caller of function code X'0003' who is not the same task as the one who called function code X'0001', will not be allowed to end cache creation. This function is further defined by the Option parameter.

If the **Option** parameter is X'0001':

- The cache is made available, via system wide name/token service, to callers of function x'0004' so they can retrieve records from the cache. Any previous cache of the same *Cache-name* is deleted.
- The cache is hardened to the RACF database if the CACHECLS class is active and if a profile exists in that class with a profile name identical to the Cache_name provided as input to R_cacheserv.

If the **Option** parameter is X'0002', discard the new cache and leave the existing cache intact. This is used if the calling application determined that something is wrong with the new cache or encountered an error while creating it.

X'0004'-Fetch information from cache.

Retrieve information from the cache. This function is further defined by the **Option** parameter.

If the **Option** parameter is X'0001' and the cache already exists, the requested (by name) record is retrieved for the caller (or is not found). If the cache does not exist yet (for example, right after IPL), a new cache can be automatically restored and populated with records which were hardened to the RACF database the last time someone called function code x'0003'. The cache is restored if the CACHECLS class is active and the cachename_ddd_nnnnn profiles containing the cache contents exist. **Data** and **Data_Length** are updated.

If the **Option** parameter is X'0002', then retrieve the version number of the existing in-storage cache. If no in-storage cache exists, retrieve the version from the database hardened copy of the cache. Do not restore the cache from the database in this case. **Version** and **Version_Length** are updated.

X'0005'-Delete the cache.

R_cacheserv

This function is further defined by the **Option** parameter.

If the **Option** parameter is X'0001', then delete the in-storage cache only.

If the **Option** parameter is X'0002', then delete the hardened database copy of the cache only.

If the **Option** parameter is X'0003', then delete both the hardened database copy and in-storage cache.

Option

The name of a half word containing an option value for the specified function code.

See *Parameter Usage* for the function codes to which the **Option** parameter applies. Valid option values and their effect on the specified function code are described above under the **Function_code** parameter.

Version

The name of a data field containing the version identifier of this cache. Length is specified in *Version_length*.

When hardening to the RACF database, the cache version identifier is compared with the version already hardened, if a version has been hardened, to avoid hardening the same cache again. If the versions are the same, the new cache will not be hardened.

Version_length

Name of a fullword containing the length of the version of the cache being created or the amount of storage the caller has provided for retrieval. Its value must be between 1 and 255. When **Function_code** X'0004' (**Fetch**) and **Option** X'0002' are specified, *Version_length* will be set to the actual length of the cache version identifier. If return and reason codes indicate that an insufficient length value was specified, resulting in a return of partial data, use the updated *Version_length* value to obtain a larger results area and resubmit the request.

Cache_name

The name of a data field containing the name of the cache. The name must be 6 bytes in length and must start with an R. The remaining 5 characters may consist of any combination of the characters A through Z, numbers 0 through 9, and the special characters @, #, and \$. Names starting with RZ are reserved for IBM use. Lower case characters will be folded to upper case.

The cachename is used internally to isolate multiple caches from each other. It is also used to generate names of the dataspace(s) that will form the cache, and also the profile names used to harden the contents of the cache to the RACF database as profiles in the CACHECLS class.

Record_name_ptr

Function codes X'0002' (**Add**) and X'0004' (**Fetch**) with **Option** X'0001' only. Name of the address of the name of the record to be added or retrieved. The length is specified in *Record_name_length*.

Record_name_length

Function codes X'0002' (**Add**) and X'0004' (**Fetch**) with **Option** X'0001' only. Name of a fullword containing the length of the name of the record to be stored or retrieved. Valid values are 1 to 8192.

Data-Ptr

Function codes X'0002' (**Add**) and X'0004' (**Fetch**) with **Option** X'0001' only.

Name of the address of data to be stored in cache (X'0002'), or name of the address where data is to be placed during retrieval (X'0004'). Length is specified in **Data_length**.

Data_length

Function codes X'0002' (**Add**) and X'0004' (**Fetch**) with **Option** X'0001' only. Name of a fullword containing the length of the data in the record to be stored or the amount of storage the caller has provided for retrieval. Valid values are 1 to 2,000,000,000.

When Function_code X'0004' (**Fetch**) and **Option** X'0001' are specified, **Data_length** will be set to the actual length of the data requested. If return and reason codes indicate that an insufficient length value was specified, resulting in a return of partial data, use the updated **Data_length** value to obtain a larger results area and resubmit the request.

Return and reason codes

IRRSCH00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful. For function code X'0003', the cache was both completed and also hardened successfully to the RACF database.
0	0	4	Function code X'0003' completed and hardened the cache successfully, but encountered problems deleting excess CACHECLS profiles from the RACF database. A record is cut to LOGREC with more diagnostic information.
0	0	8	Function code X'0003' completed the cache successfully, but the attempt to harden the cache to the RACF database failed. A record is cut to LOGREC with more diagnostic information.
0	0	12	Function code X'0003' completed the cache successfully but no attempt was made to harden the cache to the RACF database. Either the CACHECLS class was not active or the cachename profile had not been defined in the class.
4	0	0	RACF is not installed.
8	8	0	Invalid function code.
8	8	4	Parameter list error.
8	8	8	An internal error was encountered. A record may be cut to LOGREC with more diagnostic information.
8	8	12	A recovery environment could not be established.
8	8	16	Not authorized to use this service.
8	8	20	Record not found during fetch.

R_cacheserv

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
8	8	24	Record found during fetch, but Data_length is too small to fit all of the data. Data_length is set to the length needed to satisfy this request. Partial data is copied to the field pointed to by Data_ptr. If the specified Option value is X'0002', then Version_length is too small to fit all of the data. Version_length is set to the length needed to satisfy this request. Partial data is copied to the field pointed to by Version.
8	8	28	No CACHECLS profiles were found in the RACF database during the restore phase of Fetch , or the CACHECLS class is not active. No cache exists.
8	8	32	Error encountered reading CACHCLS profiles from the RACF database during the restore phase of Fetch . No cache exists. A record is cut to LOGREC with more diagnostic information
8	8	36	Only the caller of Start may Add or End . You may not add data to the cache, or End this cache.
8	8	48	This task has called Start , but has not yet called End . Fetch or Delete is not allowed by this task until End has been called.
8	8	72	Not invoked in task mode.

Parameter usage

Function	Start new cache	Add record to cache	End cache creation	Fetch cache record	Delete cache
Function Code	X'0001'	X'0002'	X'0003'	X'0004'	X'0005'
ParmALET	In	In	In	In	In
NumParms	In	In	In	In	In
Option	N/A	N/A	In	In	In
Version	In	N/A	N/A	Out	N/A
Version_length	In	N/A	N/A	In/Out	N/A
Cache_name	In	In	In	In	In
Record_name_ptr	N/A	In	N/A	In	N/A
Record_name_length	N/A	In	N/A	In	N/A
Data_ptr	N/A	In	N/A	In/Out	N/A
Data_Length	N/A	In	N/A	In/Out	N/A

Usage notes

1. An ALET must be specified for the SAF_return_code, RACF_return_code, and RACF_reason_code parameters, and a single ALET specified for all of the remaining parameters.
2. The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, a parameter containing the total number of parameters is used: NumParms. This parameter tells how many parameters appear in the list INCLUDING the number of parameters parameter. For this release, NumParms must be set to 10.
3. Use of the **Add** function code first requires an invocation of R_cacheserv with the **Start** function code. After all records have been added, R_cacheserv must be invoked one additional time with the **End** function code, to indicate that the cache has been filled and should be made available for use. Only the issuer of **Start** (same task) can **Add** and **End**.
4. To allow the R_cacheserv callable service to harden/restore the cache to/from the RACF database as profiles in the CACHECLS class, two steps must be taken:
 - a. the class must be made active by the RACF SETROPTS CLASSACT command, i.e. SETROPTS CLASSACT(CACHECLS)
 - b. a base profile for this cache must be defined in the CACHECLS class via the RACF RDEFINE command, i.e. RDEFINE CACHECLS cachename, where cachename is the Cache_name given as input to the R_cacheserv callable service.

Unless both of these steps are taken, the harden and restore phases of the **End** and **Fetch** functions, respectively, will not be performed for the cache identified by Cache_name.

5. When the cache is hardened to the RACF database, the cache contents are written to the database as profiles containing 50K pieces of the cache with the last profile's size being less than or equal to 50K. The names of the profiles are constructed from the input Cache_name parameter by adding the values '_ddd', where ddd is the sequential daspace number (in decimal), starting with '001' and '_nnnnn', where nnnnn is the number of the profile containing cache information for that daspace, also in decimal. The first 50K of the cache is written as cachename_001_00001, the second as cachename_001_00002, etc. The profiles will be created having as owner the same owner as that of the base profile.
6. If a request is made to **Start** a cache, followed by any number of **Add** requests, then **Start** is requested again for the same cache name without an intervening **End** request, this will result in the **Start** of a new empty cache, causing all records which were previously added to be discarded.
7. If a **Start**, **Add**, or **End** (Option X'0001') results in a SAF return code of 8, the state of the cache is undefined and it is highly recommended that R_cacheserv be invoked again, specifying **End** with **Option X'0002'** to discard the new cache, leaving the existing cache intact. Note that if the SAF return code 8 was caused by an ABEND during **Start** or **Add**, **End** with Option X'0002' will result in SAF return code 8 with RACF return code 8 and RACF reason code 36, indicating that the new cache was already discarded during ABEND recovery processing.
8. If more than one record is added to the cache with the same name (specified using the **Record_name_ptr** parameter), **Fetch** results are unpredictable.

R_cacheserv

9. The dataspace(s) that form the cache are associated with the Master address space and are persistent so that records in the cache can be fetched from any address space. Function code X'0005' can be used to delete the cache when it's contents no longer need to be accessed.

Related services

None

R_chaudit (IRRSCA00): Change Audit Options

Function

The **R_chaudit** service verifies that the user has authority to change the audit options for the specified file and, if so, sets the audit bits from the input audit options parameter.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF authorization

None

Format

```
CALL IRRSCA00 (Work_area,  
              ALET, SAF_return_code,  
              ALET, RACF_return_code,  
              ALET, RACF_reason_code,  
              ALET, Audit_options,  
              ALET, FSP,  
              ALET, File_identifier,  
              ALET, CRED  
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each

parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Audit_options

The name of a word containing the audit options to be set. For RACF, the following options are defined:

- Audit options can be specified for each type of access:
 - Byte 1** read access audit options
 - Byte 2** write access audit options
 - Byte 3** execute/search access audit options
 - Byte 4** audit flag
- The following flags are defined for each of the first three bytes of audit options:
 - X'00'** do not audit any access attempts
 - X'01'** audit successful access attempts
 - X'02'** audit failed access attempts
 - X'03'** audit both types of attempts
- In the last byte (the audit flag), the last bit indicates whether user audit options or auditor audit options should be set:
 - X'00'** set user audit options
 - X'01'** set auditor audit options

Reserved bits in the audit options parameter must be zero.

FSP

The name of the IFSP for the file whose audit options are to be changed.

File_Identifier

The name of a 16-byte area containing a unique identifier of the file.

CRED

The name of the CRED structure for the current file system syscall. See *z/OS Security Server RACF Data Areas*.

Return and reason codes

IRRSCA00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The user is not authorized to change the file's user audit options.

R_chaudit

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
8	8	8	The user is not authorized to change the file's auditor audit options.
8	8	12	An internal error occurred during RACF processing.
8	8	24	Reserved bits in an input parameter were not zero.
8	8	32	The CRED user type is not supported.

Usage notes

1. This service is intended only for use by a z/OS UNIX file system and by z/OS UNIX servers. The service contains support for z/OS UNIX servers, but cannot be directly invoked by a z/OS UNIX server.
2. Two sets of audit bits exist for a file, one for auditor-specified options and one for user-specified options. The audit flag in the parameter list indicates which type of options should be set.
If the audit flag indicates auditor options, the user must have auditor authority. Auditors can set the auditor options for any file, even those they do not have path access to or authority to use for any other reason.
If the audit flag indicates user options, the user must be a superuser or must be the owner of the file (that is, the effective UID of the calling process is equal to the owner UID of the file.)
3. If the change is being made for an open file, that pathname in the CRED is not used.
4. An audit record is optionally written, depending on the audit options in effect for the system.

Related Services

None

R_chmod (IRRSCF00): Change File Mode

Function

The **R_chmod** service checks whether the calling process is authorized to change the mode of the specified file (identified by the input IFSP) and, if so, changes the permission bits and the S_ISUID, S_ISGID, and S_ISVTX bits in the IFSP to the values specified by the mode parameter.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts
Locks:	No locks held

Control parameters:

The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. To change the mode, the user must be a superuser or must be the owner of the file. If the user can change the mode and the user is not a superuser, the S_ISGID bit is cleared, except when the owner z/OS UNIX group identifier (GID) of the file is equal to the effective GID or to one of the supplementary groups of the calling process.
2. Only a superuser or directory/file owner can change the S_ISVTX bit.
3. If the caller is not superuser, or the file owner, an authorization check is performed for READ access to the resource named SUPERUSER.FILESYS.CHANGEPERMS in the UNIXPRIV class. If the authorization check is successful, the caller is treated as a superuser.

Format

```
CALL IRRSCF00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Mode,
               ALET, FSP,
               ALET, File_Identifier,
               ALET, CRED
               )
```

Parameters**Work_area**

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a full word in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Mode

The name of a word containing the mode value (the file type, the permission bits, and the S_ISUID, S_ISGID, and S_ISVTX bits) to be set in the IFSP for the file.

See "File Type and File Mode Values" on page 4 for a definition of the security bits in the mode parameter. Reserved bits in the mode parameter must be zero.

R_chmod

FSP

The name of the IFSP for the file whose mode bits are to be changed.

File_Identifier

The name of a 16-byte area containing a unique identifier of the file.

CRED

The name of the CRED structure for the current file system syscall. See *z/OS Security Server RACF Data Areas*.

Return and Reason Codes

IRRSCF00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The user is not authorized to change the mode of the file.
8	8	12	An internal error occurred during RACF processing.
8	8	32	The CRED user type is not supported.

Usage Notes

1. This service is intended only for use by a z/OS UNIX file system and by z/OS UNIX servers. The service contains support for z/OS UNIX servers, but cannot be directly invoked by a z/OS UNIX server.
2. The mode word is mapped by the z/OS UNIX macro BPXYMODE.
3. If the audit function code indicates an open file, the path name in the CRED is not used.
4. An audit record is optionally written, depending on the audit options in effect for the system.

Related Services

makeFSP, R_umask, R_chown, R_setfacl, ck_access

R_chown (IRRSCO00): Change Owner and Group

Function

The **R_chown** service checks to see whether the user is authorized to change the owner of the file, and, if so, changes the owner z/OS UNIX user identifier (UID) and z/OS UNIX group identifier (GID) to the specified values.

If the user is authorized to change the file, the S_ISUID and S_ISGID bits are cleared in the IFSP.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN or PASN not = HASN

AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. This service implements the `_POSIX_CHOWN_RESTRICTED` feature in POSIX 1003.1.

If `_POSIX_CHOWN_RESTRICTED` is in effect, then:

- A user can change the owner z/OS UNIX user identifier (UID) value only if the user is a superuser.
- A user can change the owner z/OS UNIX group identifier (GID) of a file if:
 - The user is a superuser,
 - Or, all of the following are true:
 - The effective UID of the calling process is equal to the owner UID of the file (that is, the user is the owner of the file).
 - The input UID is equal to the owner UID of the file or -1
 - The input z/OS UNIX group identifier (GID) is equal to the effective GID or to one of the supplemental groups of the calling process.

If `_POSIX_CHOWN_RESTRICTED` is not in effect, then:

- A user can change the owner z/OS UNIX user identifier (UID) value AND the owner z/OS UNIX group identifier (GID) of a file if:
 - The user is a superuser
 - The effective UID of the calling process is equal to the owner UID of the file (that is, the user is the owner of the file)
2. If the caller is not superuser, an authorization check is performed on the resource name in the `UNIXPRIV` class indicated in Table 68. If the authorization check is successful, the caller is treated as a superuser.

Table 68. *UNIXPRIV* class resource names used in `R_chown`

Audit function code	Resource name	Access required
N/A	SUPERUSER.FILESYS.CHOWN	READ

Format

```
CALL IRRSC000 (Work_area,  
              ALET, SAF_return_code,  
              ALET, RACF_return_code,  
              ALET, RACF_reason_code,  
              ALET, UID,  
              ALET, GID,  
              ALET, FSP,  
              ALET, File_identifier,  
              ALET, CRED  
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

UID

The name of a word containing the z/OS UNIX user identifier (UID) to be set as the file owner UID or -1 to indicate that:

1. This field is not changed in the IFSP.
2. The z/OS UNIX group identifier (GID) can be changed.

GID

The name of a word containing the z/OS UNIX group identifier (GID) to be set as the file owner GID or -1 to indicate that this field is not changed in the IFSP.

FSP

The name of the IFSP for the file whose owner z/OS UNIX user identifier (UID) and z/OS UNIX group identifier (GID) are to be changed.

File_Identifier

The name of a 16-byte area containing a unique identifier of the file.

CRED

The name of the CRED structure for the current file system syscall. See *z/OS Security Server RACF Data Areas*.

Return and Reason Codes

IRRSCO00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The z/OS UNIX user identifier (UID) is not valid.
8	8	8	The z/OS UNIX group identifier (GID) is not valid.
8	8	12	An internal error occurred during RACF processing.
8	8	20	The user is not authorized to change the owner UID or GID.
8	8	32	The CRED user type is not supported.
8	8	36	The user is not authorized to set the specified GID.

Usage Notes

1. This service is intended only for use by a z/OS UNIX file system and by z/OS UNIX servers. The service contains support for z/OS UNIX servers, but can not be directly invoked by a z/OS UNIX server.
2. If the input UID or GID (or both) is equal to -1, that field is not changed in the IFSP.
3. If the audit function code indicates an open file, the pathname in the CRED is not used.
4. An audit record is optionally written, depending on the audit options in effect for the system.

Related Services

query_file_security_options, R_chmod, R_setfacl

R_datalib (IRRSDL00): OCSF Data Library

Function

The **R_datalib** service provides the function required to implement the Open Cryptographic Services Facility Data library functions.

Requirements

Authorization:	PSW key 8, non-APF authorized, problem state
Dispatchable unit mode:	Task
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.

Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. The words containing the ALETs must be in the primary address space.

Linkage Conventions

The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, the last word in the parameter list must have a 1 in the high-order (sign) bit.

RACF Authorization

- The following authority is required for the DataGetFirst, DataGetNext, and GetUpdateCode functions:
 1. If the RACF_user_ID field is the same as the user ID of the command issuer, then the required authority is READ to resource IRR.DIGTCERT.LISTRING in the FACILITY class.
 2. If the RACF_user_ID field is not the same as the user ID of the command issuer, then the required authority is UPDATE to the resource IRR.DIGTCERT.LISTRING in the FACILITY class.
- The CheckStatus function requires READ authority to the resource IRR.DIGTCERT.LIST in the FACILITY class.
- The DataAbortQuery function requires no authority.
- RACF SPECIAL or CONTROL authority to the resource IRR.DIGTCERT.GENCERT in the FACILITY class is required to retrieve the private keys of CERTAUTH and SITE certificates.
- The IncSerialNum function requires RACF SPECIAL or appropriate authority to the resource IRR.DIGTCERT.GENCERT in the FACILITY class, READ if the certificate is owned by the caller, CONTROL if the certificate is a SITE or CERTAUTH certificate.

Format

```
CALL IRRSDL00 (Work_area,
              ALET, SAF_return_code,
              ALET, RACF_return_code,
              ALET, RACF_reason_code,
              Function_code,
              Attributes,
              RACF_user_ID,
              Ring_name,
              Parm_list_version,
              Parmlist
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space and must be on a double word boundary.

ALET

The name of a word containing the ALET for the following parameter. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Function_code

The name of a 1 byte input area containing the function code

X'01' DataGetFirst: Locate and return the first trusted certificate in the ring specified in Ring_name, based on the selection criteria.

On a DataGetFirst function, the user may specify some selection criteria by setting Number_predicates to 1, and then supplying some attribute information, such as attribute type, and the length and address of the attribute data. The data in the returned certificate will match the attribute data supplied.

X'02' DataGetNext: Locate and return the next trusted certificate in the ring, based on the criteria specified in DataGetFirst.

X'03' DataAbortQuery: Free resources from previous DataGetFirst and DataGetNext requests.

X'04' CheckStatus: Return the TRUST/ NOTRUST status for a specified certificate.

X'05' GetUpdateCode: Return the sequence number for the ring specified. A change in the ring sequence number, (from a previously obtained ring sequence number,) indicates that the ring has changed. A ring is considered changed when the list of certificates in the ring has changed, or the digital certificate information for a certificate in the ring has changed.

X'06' IncSerialNum: Increment and return the last serial number field (CERTLSER) associated with the input certificate.

If the function code is not one of the preceding values, a parameter list error is returned.

Attributes

The name of a 4 byte area containing bit settings that direct the function to be performed. The bit settings are mapped as follows:

- Functions DataGetFirst (X'01') and DataGetNext (X'02')
x'80000000' - CDDL_ATT_ALL_KEYTYPES flag. Directs R_Datalib to differentiate between PCICC key types and ICSF key types when returning the Function Specific Parameter List field Private_key_type. When this flag is off, R_Datalib will treat either key type as an ICSF key type and return value x'00000002'.

All other bit positions are reserved and must be set to zero to ensure compatibility with future enhancements.

- All other functions

R_datalib

All bit positions are reserved and must be set to zero to ensure compatibility with future enhancements.

RACF_userid

The name of a 9 byte input area that consists of a 1 byte length field followed by up to 8 characters. It must be specified in upper case. If not specified, the length must equal 0. If not specified, the current user ID is the ring owner.

Ring_name

The name of a variable length input area that consists of a 1 byte length followed by up to 237 characters that represent the ring name. Ring_name is ignored for the CheckStatus, DataAbortQuery, IncSerialNum, and DataGetNext functions. Ring_name is used for the DataGetFirst service and the GetUpdateCode service.

parm_list_version

A four byte input value which contains the version number for the following field, parm_list. This field must be set to zero.

Parm_list

Specifies the address of the function specific parameter list for the function specified in Function_code. These are defined in "Function Specific Parameter Lists for IRRSDL00".

Return and Reason Codes

IRRSDL00 may return one of several values as the SAF and RACF return and reason codes. This section defines the return codes which can be returned by any of the functions.

Table 69. IRRSDL00 Return Codes

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	Parameter list error occurred. Attributes was not specified as 0 or the last word in the parameter list did not have the higher order bit on.
8	8	8	Not RACF authorized to use the requested service.
8	8	12	Internal error caused recovery to get control.
8	8	16	Unable to establish a recovery environment.
8	8	20	Requested Function_code not defined.
8	8	24	Parm_list_version number not supported.
8	8	28	Error in Ring_name length or RACF_userid length.
8	8	72	Caller not in task mode.

Function Specific Parameter Lists for IRRSDL00

For each function code, there is a function specific parameter list.

Function Specific Parameter Lists for DataGetFirst and DataGetNext:

Results_handle

A 4 byte address pointing to an input area. The input area must be at least 20 bytes in length, and it is that input area (not Results_handle) that is mapped as follows:

dbToken

A 4 byte value reserved for use by RACF. This value must be preserved for subsequent calls to DataGetNext and DataAbortQuery.

Number_predicates

A 4 byte integer input value. A zero, X'00000000', indicates that there are no selection criteria. This value is only used on a DataGetFirst function. A one, X'00000001', indicates that a query on a particular attribute is being performed. All other values result in a parameter error.

Attribute_ID

A 4 byte integer input value which identifies the attribute that is being queried. This field is ignored if Number_predicates is zero, X'00000000'. If Number_predicates is one, X'00000001' this field must have one of the values listed below:

X'00000001'

Attribute data to match on is label.

X'00000002'

Attribute data to match on is default flag. Attribute data supplied by Attribute_length or Attribute_ptr will either be zero, X'00000000' or non-zero.

X'00000003'

Attribute data to match on is the DER encoded subject's distinguished name.

If the Attribute_ID is not one of the preceding values, a parameter ist error is returned.

Attribute_length

A 4 byte input value containing the length of the attribute data. This field is ignored if Number_predicates is zero, X'00000000'.

Attribute_ptr

A 4 byte input address which points to the attribute data. This field is ignored if Number_predicates is zero, X'00000000'.

Certificate_Usage

A 32 bit output flag value, indicating the usage of the certificate. Certificate_Usage may have the values:

X'00000002'

Certauth

X'00000008'

Personal

X'00000000'

Other (site)

X'ffffff5'

Reserved

R_datalib

Default

A 4 byte output value. A X'00000000' value indicates that this certificate is not the default certificate for the ring. A non-zero value indicates that this is the default certificate for the ring.

Certificate_length

A 4 byte value containing the length of the certificate. On input, it contains the length of the field pointed to by certificate_ptr. On output, it contains the actual size of the certificate that is returned. A zero indicates that no certificate was returned.

Certificate_ptr

A 4 byte input value containing the address of the DER encoded certificate output area.

Private_key_length

A 4 byte value containing the length of the private key. On input, it contains the length of the field pointed to by private_key_ptr. On output, it contains the actual size of the private key that is returned. A zero indicates that no private_key was returned.

If private_key_length is zero, then private_key_bitsize and private_key_type are not returned.

Private_key_ptr

A 4 byte input value containing the address of the private key output area.

Private_key_type

A 4 byte output value indicating the form of the private key. The valid values are:

X'00000001'

PKCS #1 private key, DER encoded

X'00000002'

ICSF key token label

X'00000003'

PCICC key token label

Note: this value is returned only when the CDDL_ATT_ALL_KEYTYPES attributes flag is set. If not set, PCICC key labels are returned as ICSF key labels.

Private_key_bitsize

A 4 byte output value indicating the size of the private key, expressed in bits.

Label_length

A 4 byte value containing the length of the label. On input, it contains the length of the field pointed to by label_ptr. This field must be at least 32 bytes long. On output, it contains the actual size of the label that is returned. A zero value indicates that no label was returned.

Label_ptr

A 4 byte input value containing the address of the label output area.

CERT_user_ID

A 9 byte output area containing a 1 byte length, followed by the user ID which owns the certificate.

The 1 byte length must specify a length of 8. The user ID must be left-justified and padded with blanks.

Subjects_DN_length

A 4 byte input value containing the length of the DER encoded subject's distinguished name. On input, it contains the length of the field pointed to by Subjects_DN_ptr. On output, it contains the actual size of the subject's distinguished name that is returned.

Subjects_DN_ptr

a 4 byte input value containing the address of the subject's distinguished name output area.

Record_ID_length

On input, it contains the length of the field pointed to by Record_ID_ptr. This field must have a length of at least 246 bytes. On output, it contains the actual size of the record ID that is returned. A zero value indicates that no record ID was returned.

Record_ID_ptr

A 4 byte input value which points to a caller-provided 246 byte output area. This output area contains the record_ID returned from the callable service.

Usage Notes

1. A private key is only returned when:
 - The certificate's ring usage is personal, and
 - the caller's user ID is the user ID associated with the certificate profile or, for CERTAUTH and SITE certificates, the caller is RACF SPECIAL or has CONTROL authority to the resource IRR.DIGTCERT.GENCERT in the FACILITY class.
2. The DataAbortQuery function must be called once for each DataGetFirst call, whether or not DataGetNext calls are made between the DataGetFirst and DataAbortQuery calls. The caller must pass the same dbToken to DataAbortQuery call as was returned from the DataGetFirst call. If these conditions are not met, system resources will not be freed.

Return and Reason Codes for DataGetFirst and DataGetNext

The return codes that may be returned from the DataGetFirst and DataGetNext functions are:

Table 70. DataGetFirst and DataGetNext Return Codes

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
8	8	32	Length error in attribute_length, Record_ID_length, label_length, or CERT_user_ID.
8	8	36	dbToken error. The token may be zero, in use by another task, or may have been created by another task.
8	8	40	Internal error while validating dbToken.
8	8	44	Record not found.

Table 70. DataGetFirst and DataGetNext Return Codes (continued)

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
8	8	48	An output area is not long enough. One or more of the following input length fields were too small: Certificate_length, Private_key_length, or Subjects_DN_length. The length field(s) returned contain the amount of storage needed for the service to successfully return data.
8	8	52	Internal error while obtaining record private key data.
8	8	56	Parameter error - Number_predicates or Attribute_ID in error.
8	8	80	Internal error while obtaining ring certificate information or record trust information.
8	8	84	Profile for Ring_name not found.

Function Specific Parameter List for the DataAbortQuery Function

Results_handle

The 4 byte address value, which was returned from a prior DataGetFirst or DataGetNext request. This value is provided by the caller. The data area pointed to by Results_handle must have its fields preserved from prior DataGetFirst and DataGetNext calls.

Note that a DataAbortQuery call is required regardless of the return and reason codes from the corresponding DataGetFirst call. The DataAbortQuery function requires no authority.

Return and Reason Codes for DataAbortQuery Function: The return codes that may be returned from the DataAbortQuery function are:

Table 71. DataAbortQuery Return Codes

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
8	8	36	dbToken error. The token may be zero, in use by another task, or may have been created by another task.
8	8	40	Internal error while validating dbToken.

Function Specific Parameter List for the CheckStatus Function

Certificate_length

The 4 byte input value, containing the length of the certificate.

Certificate_ptr

The 4 byte input address value, containing the address of the DER encoded certificate.

CheckStatus Return Codes: The return codes that may be returned from the CheckStatus function are:

Table 72. CheckStatus Return Codes

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	Certificate is trusted or certificate not registered with RACF.
8	8	60	Internal error - Unable to decode certificate.
8	8	64	Certificate is registered with RACF as not trusted.
8	8	68	Parameter error - zero value specified for Certificate_length or Certificate_ptr.

Related Services

None

R_dceauth (IRRSDA00): Check a User's Authority

Function

The R_dceauth service enables an application server to check a RACF-defined user's authority to access a RACF-defined resource. It is intended to be used only by the z/OS UNIX kernel on behalf of an application server.

The client's identity can be specified by:

- The ACEE
- The *RACF_userid* parameter
- The cell and principal UUID parameter pair

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR ASC mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held

RACF Authorization

1. RACF checks the ACEE, *RACF_userid*, and the UUID parameters in the following order:
 - If the ACEE parameter has been specified, this parameter is used to identify the user for this authorization request.
 - If the ACEE parameter has not been specified, and the *RACF_userid* parameter is present, this parameter is used to identify the user for this authorization request.

R_dceauth

- If neither the ACEE nor the *RACF_userid* parameter is present, the *Cell_string_uuid* and the *Principal_string_uuid* parameters are used to identify the user for this authorization request.
- If the ACEE, *RACF_userid*, and UUID parameters have not been supplied, RACF uses the current task level ACEE if it is found. If there is no task level ACEE, RACF uses the address space ACEE, if it is present, to identify the user for this authorization request.

Format

```
CALL IRRSDA00 (Work_area,  
              ALET, SAF_return_code,  
              ALET, RACF_return_code,  
              ALET, RACF_reason_code,  
              ACEE_ptr,  
              ALET, Principal_string_uuid,  
              Cell_string_uuid,  
              RACF_userid,  
              RACF_class,  
              entity_name,  
              entity_length,  
              access_requested  
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

ACEE_ptr

The name of a fullword containing the address of an area that contains a previously created ACEE. If the caller does not specify an ACEE, this area must contain binary zeros. The ACEE parameter is **not** specified by an ALET. This parameter must be in the primary address space.

ALET

The name of a word which must be in the primary address space and which contains the ALET for the following fields:

- Principal_string_uuid
- Cell_string_uuid
- RACF_userid
- RACF_class
- Entity_name
- Entity_length
- Access_requested

Principal_string_uuid

The name of an area containing the string form of the client's DCE UUID. If the caller does not specify the client's DCE UUID, then the first character of the area must be a NULL byte. (That is, the first byte of the 36-byte area must contain X'00'.)

Cell_string_uuid

The name of an area containing the string form of the cell DCE UUID. The string form of the cell UUID, if supplied by the caller, must be 36 bytes long.

The *Cell_string_uuid* must be the name of a 36-byte area that contains one of the following:

- The string form of the cell UUID
- A null byte (X'00') as the first character of the 36-byte area. If the home cell UUID is not applicable and the caller wants to obtain cross linking information using only the DCE principal UUID, the caller must pass the *Cell_string_uuid* parameter with the first byte of this field containing a null byte of X'00'.

RACF_userid

The name of a nine-byte area, which consists of a one-byte length field followed by up to eight characters. It must be specified in uppercase. If not specified, the length must equal zero.

RACF_class

The name of an eight-byte area containing the RACF class name of the resource (such as TAPEVOL). The class name must be

- Left justified
- Padded to the right with blanks
- Specified in uppercase
- Defined to RACF via the RACF class descriptor table supplied by IBM or the installation-defined RACF class descriptor table (described in the *z/OS Security Server RACF System Programmer's Guide*).

Entity_name

The name of an area that contains the RACF resource profile name (such as TAPE01). It must be specified in uppercase.

Entity_length

The name of an area that contains the halfword length of the *entity_name*. The valid range of this parameter is 1 to 246 characters.

Access_requested

The requested access (READ, UPDATE, CONTROL, ALTER) to the resource, which is the name of a one-byte area containing:

Requested Access	Value
READ Access	X'02'
UPDATE Access	X'04'
CONTROL Access	X'08'
ALTER Access	X'80'

Return and Reason Codes

IRRSDA00 may return the following values in the reason and return code parameters:

R_dceauth

SAF return code	RACF return code	RACF reason code	Explanation
0	0	0	Access granted
4	0	0	RACF not installed, or RACF is not active
8	8	4	The resource is not defined to RACF
8	8	8	User is not authorized to access the resource
8	8	12	Internal processing error
8	8	16	Recovery environment could not be established
8	8	20	No mapping to a RACF user ID exists for the supplied UUID pair
8	8	24	Parameter list error
8	8	28	DCEUUIDS class is not active. RACF is not able to map the supplied UUIDs to a RACF user ID.
8	8	32	RACF was unable to create a security environment for the user ID specified.
8	8	36	The user ID is not RACF defined.

Usage Notes

1. This service may **not** be used to determine access to z/OS UNIX resources, such as HFS files or to data sets.
2. The DATASET class is not valid for this service.

Parameter Usage:

Parameter	Direction	Value
SAF_return_code	Output	
RACF_return_code	Output	
RACF_reason_code	Output	
ACEE_ptr	Input	Optional
Principal_string_uuid	Input	Optional
Cell_string_uuid	Input	Optional
RACF_userid	Input	Optional
RACF_class	Input	Required
entity_name	Input	Required
entity_length	Input	Required
access_requested	Input	Required

Related Services

R_dceruid

R_dceinfo (IRRSDI00): Retrieve or Set User Fields

Function

The RACF R_dceinfo callable service retrieves or sets the following fields from a user profile DCE segment:

- The DCE principal name associated with this RACF user
- DCE UUID of this user
- DCE cell name that this user is defined to (HOME CELL)
- DCE cell UUID that is associated with DCE cell that this user is defined to (HOMEUUID)
- A flag byte that indicates whether z/OS DCE creates a DCE security context (autologin) automatically

The action performed by this callable service is based on the function code passed by the caller in the R_dceinfo parameter list.

- When the function code is set to EXTRACT, R_dceinfo retrieves the information requested from the user's DCE segment.
- When the function code is set to REPLACE, R_dceinfo replaces the fields that have been specified in the parameter list.

Requirements

Authorization:	Any PSW key in: Supervisor state for REPLACE DCE fields Supervisor or problem state for EXTRACT DCE fields
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. The replace function fails if the user ID specified in the parameter has not been previously defined as a DCE RACF user.
2. Field level access checking does not occur when retrieving or replacing fields with this service.

Format

```
CALL IRRSDI00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Function_code,
               ALET, RACF_userid,
               ALET, Field_list,
               ALET, Output_area,
               ALET, Output_area_length
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a full word in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Function_code

The name of a one-byte area containing the function code:

X'01' Retrieve DCE fields

X'02' Replace DCE fields

RACF_userid

The name of a nine-byte area, containing a one byte length field, followed by a user ID up to eight characters long. It must be specified in uppercase.

Field_list

The name of an area containing the fields to be replaced or retrieved. The format of the parameter list is:

<i>Offset</i>	<i>Length</i>	<i>Description</i>
0	2	The length in bytes of the DCE field list
2	2	Total number of fields in the DCE field list
4	8	The name of the field
12	2	The length of the field data
14	variable	Field data

The ordered triplet (name of the field, length of the field, and field data) is a repeating data structure. This triplet can repeat for the total number of fields in the input `Field_list`.

- If the function code is X'01' (**retrieve DCE fields**), the caller is expected to provide the following fields in the **Field_list**:
 - The total length in bytes of the input field list
 - The total number of fields to be retrieved
 - The name of the fields to be retrieved
 - The length of the data

Note:

For the retrieve function, there is no input field data. The caller is expected to provide a length of binary zero.

The name of the field and the place holder of zero may repeat for the total number of fields to be retrieved.

- If the function code is **X'02'** (**replace DCE fields**), the caller is expected to be in supervisor state and to provide the following fields in the **Field_list**:
 - The total length in bytes of the input field list
 - The total number of fields to be retrieved
 - The name of the field to be retrieved
 - The length of the data
 - Field data

A problem-state caller is not authorized to replace DCE information.

Output_area

The name of an area that contains the fields obtained by the R_dceinfo service when the function code is **X'01'** (**Retrieve DCE fields**). The format of the output area is:

<i>Offset</i>	<i>Length</i>	<i>Description</i>
0	2	Total length of the output area of the retrieved data
2	2	Number of fields retrieved
4	8	Name of the retrieved field
12	2	Length of the retrieved field
14	variable	Field data

The ordered triplet (name of the field, length of the field, and field data) is a repeating data structure. This triplet can repeat for the number of times shown in the output_area 'number of fields retrieved' count.

Output_area_length

The name of a fullword that contains the length of the output_area that is supplied by the caller of this service.

Return and Reason Codes

IRRSDI00 may return the following values in the reason and return code parameters:

SAF return code	RACF return code	RACF reason code	Explanation
0	0	0	Service successful.
4	0	0	RACF not installed.
8	8	4	Caller is not authorized.
8	8	12	Internal error during RACF processing.
8	8	16	Recovery Environment could not be established.
8	8	20	User does not have a DCE segment.
8	8	24	Length of the output area is too small to contain the data retrieved.
8	8	28	Parameter list error.
8	8	32	User ID specified does not exist.

Usage Notes

1. If the caller is in problem state, the *RACF_userid* specified must be the same RACF user as found in either the task level ACEE or the address space level ACEE.
2. If the caller is in supervisor state, the task and address space ACEEs are not checked. Therefore, an authorized caller may extract or replace DCE segment fields for any user who has a DCE segment.
3. The retrieve function returns fields that have been previously populated. Associated with the returned fields is a length indicator. The length indicator is set to zero if a field does not exist.
4. It is the responsibility of the caller to obtain and free the output area. If the fields to be retrieved from RACF are larger than the output area, RACF fails the request and returns the actual length required in the *output_area_length* parameter.
5. The field names supplied by the caller may be:
 - UUID
 - DCENAME
 - HOMECCELL
 - HOMEUUID
 - DCEFLAGS

The field names must be supplied as 8-character fields, left justified, and padded with blanks. They must be specified in uppercase.

6. The DCEFLAGS field is a one-byte field with the following meaning:
 - Value of X'00' means that z/OS DCE does *not* attempt to sign on this user to z/OS DCE automatically
 - Value of X'01' means that z/OS DCE attempts to automatically sign on this user to z/OS DCE

Parameter Usage:

Parameter	Direction	
	GET_INFO	PUT_INFO
SAF_return_code	Output	Output
RACF_return_code	Output	Output
RACF_reason_code	Output	Output
Function_code	Input	Input
RACF_userid	Input	Input
Field_list	Input	Input
Output_area	Output	n/a
Output_area_length	Output	n/a

Related Services

R_dcekey

R_dcekey (IRRSDK00): Retrieve or Set a non-RACF Password

Function

The RACF R_dcekey callable service enables z/OS DCE to retrieve or set a DCE password (*key*) or retrieve an LDAP bind password. The three functions supported are:

- This service retrieves the DCE password from a DCE segment. The password is decrypted using the key that was stored in the user's DCE segment when the password was encrypted.
- This service sets the DCE password in a user profile DCE segment. The password is encrypted using the key stored in the DCE.PASSWORD.KEY profile in the RACF KEYSMSTR general resource class.
- This service retrieves the LDAP bind password from the PROXY segment of a general resource profile in the LDAPBIND Class or the IRR.PROXY.DEFAULTS profile in the FACILITY Class. The password is decrypted using the key that was stored in the profile's PROXY segment when the password was encrypted (for example, when the RDEFINE or RALTER PROXY(...) command was issued.)

The operation of R_dcekey is based on the function code values of **get_key**, **put_key**, and **get_ldap_pw** in the parameter list. See "Usage Notes" on page 133 for detailed information.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. For function codes **get_key** and **put_key**, the user ID specified in the RACF_entity parameter must have a DCE segment.
2. For function code **get_ldap_pw**, the LDAPBIND Class profile specified in the RACF_entity parameter must have a PROXY segment previously created through a RDEFINE or RALTER command. If the RACF_entity is not specified, the IRR.PROXY.DEFAULTS profile in the FACILITY Class must have a PROXY segment previously created through a RDEFINE or RALTER command.

R_dcekey

Format

```
CALL IRRSDK00 (Work_area,  
              ALET, SAF_return_code,  
              ALET, RACF_return_code,  
              ALET, RACF_reason_code,  
              ALET, Function_code,  
              ALET, RACF_entity,  
              ALET, key_area,  
              ALET, key_area_length  
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF use. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Function_code

The name of a one-byte area containing the function code:

X'01' get_key (retrieve current DCE key)

X'02' put_key (set DCE key)

X'03' get_ldap_pw (get the LDAP bind password from the PROXY segment)

RACF_entity

Formally called RACF_userid. The name of a 247 byte area, which consists of a one-byte length field followed by up to 246 characters. This field is to contain the RACF entity for the password being set or retrieved.

For functions get_key and put_key, this field is the RACF user ID.

For function get_ldap_pw this field is a LDAPBIND Class general resource profile name. Setting the length byte to x'00' indicates to retrieve the default LDAP bind password from the IRR.PROXY.DEFAULTS profile in the FACILITY Class.

Key_area

The name of an area containing the DCE key, preceded by a two-byte length field.

Key_area_length

The name of a fullword that contains the length of the key_area.

Return and Reason Codes

IRRSDK00 may return the following values in the reason and return code parameters:

SAF return code	RACF return code	RACF reason code	Explanation
0	0	0	Service successful.
4	0	0	RACF not installed.
8	8	4	Entity not defined to RACF
8	8	12	Internal error during RACF processing
8	8	16	Recovery environment could not be established.
8	8	20	Entity missing required segment (DCE or PROXY)
8	8	24	No DCE key or LDAP bind password exists for the entity
8	8	28	The key_area supplied by the caller is too small
8	8	32	Parameter list error
8	8	36	RACF KEYSMSTR inactive, or the profile DCE.PASSWORD.KEY profile was not defined to the RACF KEYSMSTR with a SSIGNON segment
8	8	40	Invalid data was found in SSIGNON segment of the DCE.PASSWORD.KEY profile in the RACF KEYSMSTR
8	8	44	RACF was unable to retrieve or update the master key/master key token in the SSIGNON segment of the DCE.PASSWORD.KEY profile in the RACF KEYSMSTR
8	8	48	Unexpected error returned from RACROUTE which may be caused by specifying the entity incorrectly. Symptom record written
8	8	52	RACF cannot locate the CCA support routine
8	8	56	The invocation of the CCA support routine has failed

Usage Notes

1. When the function is `get_key`, this service returns the current DCE key in clear text form to the output area supplied by the caller. This is the value defined by the `key_area` parameter. The length of the `key_area` is supplied by the `key_area_length` parameter.
2. If the `key_area` supplied by the caller is too small to contain the user's current DCE key or the profile's LDAP password, the service sets the required length in the `key_area_length` parameter.
3. When the function is `put_key`, this service replaces the current DCE key in the specified user profile DCE segment with the value specified in the `key_area` parameter.
4. When the function is `get_ldap_pw`, this service returns the LDAP bind password in clear text form to the output area supplied by the caller. This is the

R_dcekey

value defined by the key_area parameter. The length of the key_area is supplied by the key_area_length parameter.

Parameter Usage:

Parameter	Direction		
	GET_KEY	PUT_KEY	GET_LADP_PW
SAF_return_code	Output	Output	Output
RACF_return_code	Output	Output	Output
RACF_reason_code	Output	Output	Output
Function_code	Input	Input	Input
RACF_entity	Input	Input	Input
Key_area	Output	Input	Output
Key_area_length	Input/Output	n/a	Output

Related Services

R_dceinfo

R_dceruid (IRRSUD00): Determine the ID of a Client

Function

The **R_dceruid** service enables z/OS DCE servers to determine the RACF user ID of the client from the string forms of the client's DCE UUID pair, which consists of the *home cell* UUID and the *principal* UUID. It also enables the servers to determine the DCE UUIDs of a client from the RACF user ID.

Note that this service can *only* convert a DCE UUID to a RACF user ID and a RACF user ID to a DCE UUID for users who have:

- A populated DCE segment associated with their user profile
- A DCEUUIDS-class profile that defines the association between the DCE UUIDs and the RACF user ID

The R_dceruid service is sensitive to the profiles defined to the RACF DCEUUIDS class.

To resolve a conversion request:

- If a caller specifies a UUID pair on this service's invocation to convert the UUID pair to the corresponding RACF user ID, the service searches the RACF DCEUUIDS-class profiles defined to RACF with that UUID pair.
- If a caller specifies only a principal UUID on the service's invocation to convert the principal UUID to the corresponding RACF user ID, the service searches the RACF DCEUUIDS-class profiles defined to RACF with only that principal UUID.

Requirements

Authorization:	Any PSW key in supervisor or problem state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any

ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have a FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. This service can only translate a RACF user ID to a DCE UUID and a DCE UUID to a RACF user ID for users who have:
 - A populated DCE segment associated with the user profile
 - A DCEUUIDS-class profile defined to RACF that associates a DCE UUID pair with a RACF user ID
2. Use of the R_dceruid service is authorized by the profile IRR.RDCERUID in the RACF FACILITY class. The user ID of the application server (the RACF identity associated with the application server), or a group to which the server is connected, must be permitted with READ access to the profile IRR.RDCERUID in the RACF FACILITY class. Assigning a UACC of READ to the profile IRR.RDCERUID is not recommended.

Format

```
CALL IRRSUD00 (Work_area,
               ALET,SAF_return_code,
               ALET,RACF_return_code,
               ALET,RACF_reason_code,
               ALET,Function_code,
               ALET,Principal_string_uuid,
               ALET,Cell_string_uuid,
               ALET,RACF_userid
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Function_code

The name of a one-byte area containing the function code:

R_dceruid

X'01' Return RACF user ID

X'02' Return DCE UUIDs

Principal_string_uuid

The name of an area containing the string form of the principal DCE UUID. The area must be 36 characters long.

- If the function code is **return RACF user ID**, this area must contain the principal DCE UUID as input.
- If the function code is **return DCE UUIDs**, this area is used as an output area when the principal DCE UUID is returned.

Cell_string_uuid

The name of an area containing the string form of the cell DCE UUID. The string form of the cell UUID, if supplied by the caller, must be 36 bytes long.

- If the function code is return RACF user ID, the *Cell_string_uuid* must be the name of a 36-byte area that contains one of the following:
 - The string form of the cell UUID
 - A null byte (X'00') as the first character of the 36-byte area. If the home cell UUID is not applicable and the caller wants to obtain cross linking information using only the DCE principal UUID, the caller must pass the *Cell_string_uuid* parameter with the first byte of this field containing a null byte of X'00'.
- If the function code is return DCE UUIDS, this area is used as an output area when the home cell UUID is returned.

RACF_userid

The name of a nine-byte area, which contains a one-byte length field followed by up to 8 characters. It must be specified in uppercase.

When using this callable service to return the RACF user ID associated with a DCE UUID, if the APPLDATA field in the appropriate DCEUUIDS class profile has not been populated with a RACF user ID, the service returns a 0 in the one-byte length field.

Return and Reason Codes

IRRSUD00 may return the following values in the reason and return code parameters:

SAF return code	RACF return code	RACF reason code	Explanation
0	0	0	Service successful
4	0	0	RACF not installed, or RACF is not active
8	8	0	RACF user ID specified does not exist.
8	8	4	No mapping to a RACF user ID exists for this UUID.
8	8	8	Not authorized to use this service.
8	8	12	Internal error during RACF processing
8	8	16	Recovery environment could not be established.
8	8	20	Local cell DCE UUID could not be determined for this RACF to DCE UUID conversion request.

SAF return code	RACF return code	RACF reason code	Explanation
8	8	24	The RACF DCEUUIDS class is not active. UUID to RACF conversion request could not be performed.
8	8	28	Parameter list error.
8	8	32	No mapping to a UUID exists for this RACF user ID.

Usage Notes

- This callable service allows z/OS DCE servers to associate a DCE UUID pair with a RACF user ID.
 - If the function code is **return RACF user ID**, the **R_dceruid** service returns the RACF user ID associated with the string form of the DCE UUIDs supplied by the caller, if it is available.
 - If the function code is **return DCE UUID**, the **R_dceruid** service returns the string form of the DCE UUIDs associated with the RACF user ID supplied by the caller, if it is available. RACF stores the UUIDs as uppercase characters and therefore returns the UUIDs in uppercase.

Parameter Usage:

Parameter	Direction	
	UUID to RACF user ID	RACF user ID to UUID
SAF_return_code	Output	Output
RACF_return_code	Output	Output
RACF_reason_code	Output	Output
Function_code	Input	Input
Principal_string_uuid	Input	Output
Cell_string_uuid	Input	Output
RACF_userid	Output	Input

Related Services

R_dceinfo, R_usermap

R_exec (IRRSEX00): Set Effective and Saved UIDs/GIDs

Function

The **R_exec** service sets the effective and saved z/OS UNIX user identifiers (UIDs) and z/OS UNIX group identifiers (GIDs) for a process to the specified input values. Input flags indicate whether the UIDs or GIDs or both should be changed.

R_exec returns the new values of the real, effective, and saved UIDs and GIDs in the output areas provided.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

None

Format

```
CALL IRRSEX00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Flags,
               ALET, UID,
               ALET, GID,
               ALET, UID_output_area,
               ALET, GID_output_area
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Flags

The name of a byte containing the settings of the SETUID and SETGID flags for the file being executed. The flags are:

- X'01' - SETUID
- X'02' - SETGID

- X'03' - Both SETUID and SETGID

UID

The name of a fullword containing the z/OS UNIX user identifier (UID) to be set. The UID must be defined to RACF.

GID

The name of a fullword containing the z/OS UNIX group identifier (GID) to be set. The GID must be defined to RACF.

UID_output_area

The name of a 3-word area in which the new real, effective, and saved z/OS UNIX user identifiers (UIDs), in that order, are returned.

GID_output_area

The name of a 3-word area in which the new real, effective, and saved z/OS UNIX group identifiers (GIDs), in that order, are returned.

Return and Reason Codes

IRRSEX00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The z/OS UNIX user identifier (UID) is not defined to RACF.
8	8	8	The z/OS UNIX group identifier (GID) is not defined to RACF.
8	8	12	An internal error occurred during RACF processing.
8	8	16	Recovery could not be established.

Usage Notes

1. This service is intended only for use by the MVS BCP.
2. An audit record is optionally written, depending on the options in effect for the system.
3. This service uses task level support when z/OS UNIX has indicated in the task's ACEE that this is a task level process.

Related Services

None

R_fork (IRRSFK00): Fork a Process

Function

When called from the parent process, the **R_fork** service returns the address, subpool, and key of the storage containing the user security information for the calling process.

R_fork

When called from the child process, the **R_fork** service returns the address of an area containing a copy of the security information pointed to on the initial call. The storage pointed to by the address is obtained by the subpool and key returned on the previous call.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	None. Caller handles recovery.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

None

Format

```
CALL IRRSFK00 (Work_area,  
ALET, SAF_return_code,  
ALET, RACF_return_code,  
ALET, RACF_reason_code,  
ALET, Flag,  
ALET, Data_key,  
ALET, Data_address,  
ALET, Data_length,  
ALET, Data_subpool  
)
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Flag

The name of a fullword flag that describes whether fork processing is being done in the parent's or child's address space:

X'00000000'

Fork processing is being done in parent's address space.

X'00000001'

Fork processing is being done in child's address space.

X'00000002'

Fork processing is being done in parent's address space for additional security information..

X'00000003'

Fork processing is being done in child's address space for additional security information..

Data_key

The name of a word in which:

- The storage key of the parent's USP or additional security information is returned by IRRSFK00 during parent fork processing.
- The storage key of the parent's USP or additional security information is supplied to IRRSFK00 during child fork processing.

Data_address

The name of a word in which the address of:

- The parent's USP or additional security information is returned by IRRSFK00 during parent fork processing.
- A copy of the parent's USP or additional security information is supplied to IRRSFK00 during child fork processing.

Data_length

The name of a word that contains the length of the data addressed by Data_address.

Data_subpool

The name of a word in which:

- The storage subpool for the parent's USP or additional security information is returned by IRRSFK00 during parent fork processing.
- The storage subpool for the parent's USP or additional security information is supplied to IRRSFK00 during child fork processing.

Return and Reason Codes

IRRSFK00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
0	0	4	Additional security information available.

Usage Notes

1. This service is intended only for use by the MVS BCP.
2. The key is meaningful only when the following subpools are used:
 - 129-132

R_fork

- 227–231
 - 241
3. The following user security information is propagated from the parent address space to the child address space:
 - The user security packet (USP)
 - The privilege bit (ACEEPRIV) from the parent's ACEE
 - The privilege bit (TOKPRIV) and the trusted bit (TOKTRST) from the parent's TOKEN
 - Additional security information. For RACF, this includes:
 - Controlled status
 - Keep-controlled indicators
 - Saved messages
 - RACF reason code 4 indicates that there is additional security information related to the parent address space that should be propagated to the child address space. If a reason code 4 is received when a flag value of 0 was passed, R_fork should be called again with a flag value of 2 specified. If a reason code 4 is received when a flag value of 1 was passed, R_fork should be called again with a flag value of 3 specified.
 4. This service uses task level support when z/OS UNIX has indicated in the task's ACEE that this is a task level process.

Related Services

None

R_getgroups (IRRSSG00): Get/Set Supplemental Groups

Function

The **R_getgroups** service checks the high-order bit of the input group count. See **Group_count** under "Parameters" on page 143 for more information.

The GIDs are not explicitly added to or deleted from the supplemental group list. A GID is in this list if the user was a member of the group when the user's ACEE was created through a RACROUTE REQUEST=VERIFY request and if the GID was assigned to the group before the **initUSP** service was performed for the process.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

None

Format

```
CALL IRRSGG00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, User_key,
               ALET, Group_count,
               ALET, Grouplist,
               ALET, Number_of_GIDs
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

User_key

The name of a byte containing the user's key. This key is used to store into the output grouplist area. The key is in the four high-order bits of the byte.

Group_count

The name of a word containing the number of GID entries that can be stored in the *Grouplist* area. IRRSGG00 uses the high-order bit to determine how to process the value in the parameters.

If the high-order bit of the input *Group_count* is:

1. On, the caller must store into this area the list of GIDs of the supplemental groups to be set as the supplemental groups of the current process.
2. Off, IRRSGG00 checks the input *Group_count* value. If it is:
 - a. 0, the *Grouplist* area is not used. IRRSGG00 returns the total supplemental GID count of the current process in the *Number_of_GIDs* parameter.
 - b. Less than the total supplemental GID count:
 - 1) An error code is returned.
 - 2) The GIDs of the supplemental groups for the current process are put into the *Grouplist* area, which can only accommodate the number of GIDs specified in the *Group_count* parameter.
 - 3) The count of the supplemental GIDs actually placed in the *Grouplist* area is returned in the *Number_of_GIDs* parameter.

R_getgroups

- 4) The *Group_count* field is set to the total supplemental GID count of the current process.
The supplemental groups in the *Grouplist* area are listed in the same order as the group connections shown in the output of the LISTUSER command.
- c. Greater than or equal to the total supplemental GID count:
 - 1) The GIDs of the supplemental groups for the current process are put into the *Grouplist* area.
 - 2) The supplemental GID count of the current process is put into the *Number_of_GIDs* parameter.

Grouplist

The name of an area in which the GIDs of the supplemental groups for a process are returned. The *Group_count* parameter indicates the number of entries this area can contain. The GIDs are returned as consecutive 4-byte entries.

Number_of_GIDs

The name of a word in which the number of GIDs put in the *Grouplist* area is returned.

Return and Reason Codes

IRRSGE00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	<i>Group_count</i> is less than the number of supplemental groups (see item 2b on page 143 under the Group_count parameter).
8	8	8	The grouplist address is not valid.
8	8	12	An internal error occurred during RACF processing.

Usage Note

- This service is intended only for use by the MVS BCP and by z/OS UNIX servers. The service contains support for z/OS UNIX servers, but cannot be directly invoked by a z/OS UNIX server.

Related Services

None

R_getgroupsbyname (IRRSUG00): Get Groups by Name

Function

The **R_getgroupsbyname** service checks the input group count and, if it is zero, returns the number of supplemental groups for the specified user ID. If the input count is not zero and it is less than the number of groups, an error code is

returned. If the count is not less than the number of groups, the GIDs of the supplemental groups for the specified user ID are put into the grouplist area, and the number of GIDs is returned.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. A RACF user can be connected to more than NGROUPS_MAX groups, but only up to the first NGROUPS_MAX z/OS UNIX groups will be associated with the user for this service.

The first NGROUPS_MAX z/OS UNIX groups to which a user is connected, as shown by a LISTUSER command, are the groups that get associated with the user.

Format

```
CALL IRRSUG00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, User_key,
               ALET, Userid_length,
               ALET, Userid,
               ALET, Group_count,
               ALET, Grouplist,
               ALET, Number_of_GIDs
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

R_getgroupsbyname

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

User_key

The name of a byte containing the user's key. This key is used to store into the output grouplist area and number_of GIDs word. The key is in the four high-order bits of the byte.

Userid_length

The name of a byte containing the length of the user ID.

Userid

The name of an 8-byte area containing the user ID whose groups are to be returned. The user ID must be left-justified in the area. The userid_length parameter specifies the actual length of the name.

Group_count

The name of a fullword containing the number of GID entries in the input grouplist area.

Grouplist

The name of an area in which the GIDs of the supplemental groups are returned. The GIDs are returned as consecutive 4-byte entries. The group_count parameter indicates the number of entries this area can contain.

Number_of_GIDs

The name of a word in which the number of GIDs actually put in the grouplist area is returned.

Return and Reason Codes

IRRSUG00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The group count is less than number of supplemental groups.
8	8	8	The grouplist address is not valid.
8	8	12	An internal error occurred during RACF processing.
8	8	16	Recovery could not be established.
8	8	20	RACROUTE VERIFY processing failed.
8	8	24	The user ID is not defined to RACF.

Usage Notes

1. This service is intended only for use by the MVS BCP.

Related Services

None

R_IPC_ctl (IRRSCI00): Perform IPC Control

Function

The **R_IPC_ctl** service performs a function based on the function code value in the parameter list.

- When the function is Check Owner for Remove ID, **R_IPC_ctl** checks whether the current process has the appropriate authority to the IISP.
- When the function is Check Owner and Set, **R_IPC_ctl** sets the owner's UID and GID and the mode permission bits if the current process has the appropriate authority.
- When the function is Check Superuser and Set, **R_IPC_ctl** sets the owner's UID and GID and the mode permission bits if the current process is a superuser.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user/any task if system user type is specified
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. The access checks performed are defined in XPG4 System Interfaces and Headers under msgctl, semctl, and shmctl interfaces for commands IPC_SET and the IPC_RMID. The access checks are as follows:
 - a. The effective z/OS UNIX user identifier (UID) for the calling process is used for all access checks.
 - b. If the CREI user type is system, IRRSCI00 grants authorization when the function is Check Owner for Remove ID, or updates the IPCP when the function is either Check Owner and Set or Check Superuser and Set.
 - c. The user is considered a superuser if the effective UID is zero or if the ACEE indicates trusted or privileged authority.
 - d. If the function is Check Owner for Remove ID, the user must be either a superuser or the effective UID of the process must match either the owner's UID or creator's UID in the IISP for a successful completion. Otherwise, the user is not authorized.

Note: If the caller is unauthorized as stated above, an authorization check is performed on the resource name in the UNIXPRIV class indicated in Table 73 on page 148. If the authorization check is successful, the caller is treated as a superuser.

Table 73. UNIXPRIV class resource names used in R_IPC_ctl

Function code	Resource name	Access required
1-Check Owner for Remove ID	SUPERUSER.IPC.RMID	READ

2. If the function is Check Superuser and Set, the user must be a superuser in order to set the owner's z/OS UNIX user identifier (UID), owner's z/OS UNIX group identifier (GID), and mode fields from the input parameters into the IISP for a successful completion. Otherwise, the user is not authorized.
3. If the function is Check Owner and Set, the user must be either a superuser or the effective UID of the process must match either the owner's UID or creator's UID in the IISP in order to set the owner's UID, owner's GID, and mode fields from the input parameters into the IISP for a successful completion. Otherwise, the user is not authorized.

Format

```
CALL IRRSCI00 (Work_area,
              ALET, SAF_return_code,
              ALET, RACF_return_code,
              ALET, RACF_reason_code,
              ALET, Function_code,
              ALET, Owner_UID,
              ALET, Owner_GID,
              ALET, Mode_Permissions,
              ALET, ISP,
              ALET, CREDIPC
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Function_code

The name of a 1-byte area containing the function code:

X'01' Check Owner for Remove ID

X'02' Check Owner and Set

X'03' Check Superuser and Set

Owner_UID

The name of a 4-byte area containing the new owner's UID to be set.

Owner_GID

The name of a 4-byte area containing the new owner's GID to be set.

Mode_Permissions

The name of a 4-byte area containing the new mode permission bits to be set. The following is a list of defined permission bits mapped by BPXYMODE:

S_IRUSR

Permits the process that owns the IPC member to read it.

S_IWUSR

Permits the process that owns the IPC member to alter it.

S_IRGRP

Permits the group associated with the IPC member to read it.

S_IWGRP

Permits the group associated with the IPC member to alter it.

S_IROTH

Permits others to read the IPC member.

S_IWOTH

Permits others to alter the IPC member.

Alter and write have the same meaning for access checks. Alter applies to semaphores, and write applies to message queueing and shared memory segments.

ISP

The name of the IISP for the file being accessed.

CREDIPC

The name of the CREI structure for the current IPC system callable service. The CREI contains the IPC identifier and the IPC key. See *z/OS Security Server RACF Data Areas*.

Return and Reason Codes

IRRSCI00 may return the following values in the reason and return code parameters:

SAF return code	RACF return code	RACF reason code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The user is not authorized.
8	8	12	An internal error occurred during RACF processing.
8	8	32	The CREI user type is not supported.

Usage Notes

1. This service is intended for use only by the MVS BCP.
2. An audit record is optionally written, depending on the audit options in effect for the system.
3. This service uses task level support when z/OS UNIX has indicated in the task's ACEE that this is a task level process.

Related Services

makeISP, ck_IPC_access

R_kerbinfo (IRRSMK00): Retrieve or Set Security Server Network Authentication Service Fields

Function

The **R_kerbinfo** callable service can be used to either retrieve RACF Network Authentication Service information, or to update the count of unsuccessful attempts to use a Network Authentication Service key.

The action performed by this callable service is based on the function code passed by the caller in the R_kerbinfo parameter list:

- When the function code is set to X'01', R_kerbinfo retrieves local Network Authentication Service principal information. The caller must identify the principal by providing a RACF name or a Network Authentication Service principal name.
- When the function code is set to X'02', R_kerbinfo increments the count of invalid attempts by a Network Authentication Service principal to use a key. The caller must identify the principal by providing a Network Authentication Service principal name.
- When the function code is set to X'03', R_kerbinfo resets the count of invalid attempts by a Network Authentication Service principal to use a key to zero. The caller must identify the principal by providing a Network Authentication Service principal name.
- When the function code is set to X'04', R_kerbinfo retrieves Network Authentication Service realm information. The caller may identify the realm by providing a Network Authentication Service realm profile name, or by providing a NULL name, in which case RACF will return information about the local realm, KERBDFLT.

Field data is returned to the invoker in a data structure containing a set of repeating ordered triplets, which are of the format name of the field, length of the field, and field data.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Any task
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary only
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. The words containing the ALETs must be in the primary address space.

Linkage Conventions

The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, the last word in the parameter list must have a one in the high-order (sign) bit.

Format

```
CALL IRRSMK00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               Function_code,
               RACF_name,
               KERB_name,
               Data_area,
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a full word in which the SAF router returns the SAF return code.

RACF_return_code

The name of a full word in which the service routine stores the return code.

RACF_reason_code

The name of a full word in which the service routine stores the reason code.

Function_code

The name of a one-byte area in the primary address space containing the function code:

X'01'

Retrieve local Network Authentication Service principal information.

X'02'

Increment a Network Authentication Service principal's count of invalid key attempts.

RACF will process this request much the same as it does when an invalid password is supplied during TSO logon. When the number of attempts exceeds the number of incorrect password attempts which RACF allows, set using the SETROPTS command PASSWORD REVOKE suboperand, the RACF user ID will be revoked, an ICH408I message will be issued and an SMF Type 80 record will be written.

X'03'

Reset a Network Authentication Service principal's count of invalid key attempts to zero.

RACF will process this request much the same as a successful TSO logon request and will update the date/time of the last successful logon (the LJDATE/LJTIME fields in the RACF user profile) to the current date/time.

R_kerbinfo

X'04'

Retrieve Network Authentication Service realm information.

RACF_name

The name of a 9-byte area in the primary address space consisting of a one-byte length field followed by up to 8 characters. If a value is specified for RACF_name, it must be defined RACF user ID and specified in uppercase. If a RACF user ID is not specified, the length must equal zero.

RACF_name may only be specified with function code X'01' and will be used to identify a local Network Authentication Service principal by the principal's RACF user identity.

KERB_name

The name of an area in the primary address space containing the 240 byte Network Authentication Service name. The name must be left-justified and padded with blanks. The only way to get the local realm information is to specify a null in the KERB_name field. If a Network Authentication Service realm profile name is specified, it must be realm qualified (for example, follow the DCE-like convention of /.../REALM_A/KRBTGT/REALM_B) and must be folded to all uppercase.

Note: Local Network Authentication Service principal names and realm names returned in the Data_area NAME field will not be realm qualified and will be case sensitive.

If the caller does not wish to specify a KERB_name, then the first character of the area must be a NULL byte (that is, the first byte of the 240 byte area must contain X'00').

KERB_name may be specified with any function code. For function codes X'01', X'02', and X'03', it is used to identify a local Network Authentication Service principal. For function X'04', it is used to identify a Network Authentication Service realm profile name.

Data_area

The name of an area in the primary address space for fields to be retrieved. The format of the Data_area structure is:

<i>Offset</i>	<i>Length</i>	<i>Description</i>
0	2	The length in bytes of the entire Data_area structure
2	2	Total number of fields
4	8	The name of the field
12	2	The length of the field data
14	variable	Field data

The ordered triplet (name of the field, length of the field, and field data) is a repeating data structure. This triplet will repeat for the total number of fields in the input Data_area.

The following table lists the fields that will be returned for function code X'01' (retrieve local principal information) and X'04' (retrieve realm information). The caller supplied Data_area structure must be allocated large enough for all of the fields associated with the function to be returned. The fields that will be returned, along with each field's maximum length in bytes and the order that they will be returned, can be found in the following table:

Field Name	Maximum Length	Data Type	Description
USERID	8	Character	RACF userid (N/A for function X'04')
REVOKED	1	Boolean	Flag indicating user has been revoked (N/A for function X'04')
EXPIRED	1	Boolean	Flag indicating user's password has expired (N/A for function X'04')
NAME	240	Character	Kerberos name
MINTKTLF	4	Integer	Minimum ticket life value (N/A for function X'01')
MAXTKTLF	4	Integer	Maximum ticket life value
DEFTKTLF	4	Integer	Default ticket life value (N/A for function X'01')
SALT	240	Character	Current key salt
ENCTYPE	4	Binary	Specifies the encryption types allowed for this profile. See Table 74 on page 154 for ENCTYPE values.
CURKEYV	1	Integer	Current key version (1-255)
CURKEY	10 or new format	Character	Current key - returned as 2 byte length, followed by key value (8 bytes)
PREVKEYV	1	Integer	Previous key version (1-255)
PREVKEY	10 or new format	Character	Previous key - returned as 2 byte length, followed by key value (8 bytes)

The minimum length of the Data_area structure is then 662 bytes, accounting for the two 2-byte length fields at the beginning and 13 sets of field triplets, each with the above length requirements.

The value fields CURKEY and PREVKEY returned for information retrieval function codes X'01' and X'04' have a new format. If the SETROPTS KERBLVL setting has the value of 1, the data portion of the fields of these names is in the following format:

Offset	Length	Description
0	2	The length of DES key
2	8	DES key
10	2	The length of DES3 key
12	24	DES3 key
36	2	The length of DES key with derivation
38	8	DES key with derivation

During the profile's transition from the old format of key to the new format, you may see the following:

- A profile that has both CURKEY and PREVKEY of the format Length DES key/DES key. This would occur when the SETROPTS KERBLVL(1) has been issued, but no password change has occurred since for the specific profile.

R_kerbinf

- A profile that has CURKEY in the new format and PREVKEY in the old format. This would occur when the SETROPTS KERBLVL(1) has been issued and one password change has occurred since.

The value of the ENCTYPE field will be defined as follows:

Table 74. ENCTYPE field value

Value of ENCTYPE Field (by bit position)	Encryption Type Enabled
'00000000'X	None
'000000001'X	DES
'000000002'X	DES3
'000000004'X	DES with derivation

All keys will be returned regardless of the setting of ENCTYPE.

Note: In the case where the ENCTYPE was set, a SETROPTS command is issued to set the KERBLVL back to 0, which is the original level of support, from a higher setting. The value of ENCTYPE may be unpredictable depending on when the key was generated.

The minimum length of the data_area structure passed in to the service for the retrieval function codes has been increased from 662 to 682 to account for the additional key data returned.

Return and Reason Codes

IRRSMK00 may return the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
4	4	0	User revoked prior to call.
4	4	4	User revoked by this call.
8	8	0	Invalid function code.
8	8	4	Parameter list error.
8	8	8	Internal error during RACF processing.
8	8	12	Recovery environment could not be established.
8	8	20	RACF profile does not have required segments.
8	8	24	Length of the output area is too small to contain the data retrieved.
8	8	32	RACF profile specified does not exist.
8	8	36	Mapping to RACF profile failed.

Usage Notes

1. The caller is in supervisor state, so the task and address space ACEEs are not checked. Therefore, for example, an authorized caller may extract KERB segment fields, or update the invalid key count, for any user who has a KERB segment.
2. This service returns fields that have been previously populated. Associated with the returned fields is a length indicator. The length indicator is set to zero if a field does not exist.
3. If RACF_name and KERB_name are both provided for function X'01', R_kerbinfo will use RACF_name.
4. If RACF_name is provided for any function other than X'01', a parameter list error will be returned.
5. If KERB_name is not supplied on a function X'04' request (the first character is NULL), information about the local z/OS Kerberos Security Server, KERBDFLT, will be returned. Alternatively, KERB_name may be explicitly set to KERBDFLT.
6. It is the responsibility of the caller to obtain and free the Data_area. If the fields to be retrieved from RACF are larger than the Data_area, RACF fails the request.
7. Field level access checking does not occur when retrieving fields with this service.
8. Field names are returned as 8-character fields, left-justified, and padded with blanks. They are specified in uppercase.
9. Fields that are not applicable for a function code, such as USERID for function code X'04', will be returned with the length set to zero.
10. If function code X'02' causes a user to be revoked, an ICH408I message will be issued and an SMF Type 80 record will be cut.

Parameter Usage

Table 75. Parameter Usage

Parameter	Function X'01'	Function X'02'	Function X'03'	Function X'04'
SAF_return_code	Output	Output	Output	Output
RACF_return_code	Output	Output	Output	Output
RACF_reason_code	Output	Output	Output	Output
Function_code	Input	Input	Input	Input
KERB_name	Input	Input	Input	Input
Data_area	Input/Output	N/A	N/A	Input/Output

Related Services

R_ticketserv, R_usermap

R_PKIServ (IRRSPX00): Request Public Key Infrastructure (PKI) Services

Function

The R_PKIServ SAF callable service allows applications to request the generation, retrieval, and administration of X.509 V.3 certificates and certificate requests.

Requirements

Authorization:	Any PSW key in supervisor or problem state
Dispatchable unit mode:	Task of user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have a FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. The words containing the ALETs must be in the primary address space.

RACF Authorization

The RACF authorization mechanism for this callable service varies depending on the type of function requested (end user versus administrative) and the requested provider (SAF versus PKI Services).

For the end user functions, this interface is protected by FACILITY class profiles of the form IRR.RPKISERV.*(function)*, where *(function)* is one of the end user function names described under **Function_code** below. The user ID (from the ACEE associated with the address space) for the application, is used to determine access:

NONE

Access is denied.

READ

Access is permitted based on subsequent access checks against the caller's user ID. To determine the caller, the current TCB is checked for an ACEE. If one is found, the authority of that user is checked. If there is no ACEE associated with the current TCB, the ACEE associated with the address space is used to locate the user ID.

UPDATE

Access is permitted based on subsequent access checks against the application's user ID.

ALTER OR CONTROL (or user ID is RACF SPECIAL)

Access is permitted with no subsequent access checks made.

For SAF GENCERT and EXPORT requests where the application has READ and UPDATE access, subsequent access checks are performed against the IRR.DIGTCERT.*(function)* FACILITY profiles. These are identical to the checks made by the RACDCERT TSO command. See the *z/OS Security Server RACF Command Language Reference, SC28-1919-07* and the *z/OS Security Server RACF Security Administrator's Guide, SC28-1915-07* for more information.

For PKI Services GENCERT, REQCERT, EXPORT, VERIFY, GENRENEW, and REQRENEW requests where the application has READ and UPDATE access, subsequent access checks are performed against the IRR.DIGTCERT.<*function*> FACILITY profiles as follows:

- GENCERT — This function is used to request an auto-approved certificate. The access check user ID needs to have CONTROL access to IRR.DIGTCERT.GENCERT. The access check user ID also needs appropriate

access to IRR.DIGTCERT.ADD, UPDATE access if any HostIdMapping information is specified in the certificate request parameter list or the Userid field in the certificate request parameter list indicates the certificate is being requested for another user other than the caller, otherwise READ access.

- REQCERT — This function is used to request a certificate which must be approved by an administrator before being created. The access check user ID needs to have READ access to IRR.DIGTCERT.REQCERT
- EXPORT — This function is used to retrieve (export) a certificate that was requested previously. The access check user ID needs to have appropriate access to IRR.DIGTCERT.EXPORT, UPDATE access if no pass phrase is specified on the call, READ access if a pass phrase is specified.
- VERIFY — This function is used to confirm that a given user certificate was issued by this CA and if so, return the certificate fields. The access check user ID needs to have READ access to IRR.DIGTCERT.VERIFY. It is assumed that the calling application has already verified that the end user possesses the private key that correlates to the input certificate.
- REVOKE — This function is used to revoke a certificate that was previously issued. The access check user ID needs to have READ access to IRR.DIGTCERT.REVOKE. It is assumed that the calling application has already verified the target certificate using the VERIFY function.
- GENRENEW — This function is used to generate a renewal certificate. The request submitted is automatically approved. The access check user ID needs to have READ access to IRR.DIGTCERT.GENRENEW and CONTROL access to IRR.DIGTCERT.GENCERT. It is assumed that the calling application has already verified the input certificate using the VERIFY function.
- REQRENEW — This function is used to request certificate renewal. The request submitted needs to be approved by the administrator before the certificate is renewed. The access check user ID needs to have READ access to IRR.DIGTCERT.REQRENEW. It is assumed that the calling application has already verified the input certificate using the VERIFY function.

For the administrative functions, this interface is protected by a single FACILITY class profile, IRR.RPKISERV.PKIADMIN. If the caller is not RACF SPECIAL, then the caller will need READ access to perform read operations (QUERYREQS, QUERYCERTS, REQDETAILS, and CERTDETAILS) and UPDATE access for the action operations, (MODIFYREQS and MODIFYCERTS). To determine the caller, the current TCB is checked for an ACEE. If one is found, the authority of that user is checked. If there is no ACEE associated with the current TCB, the ACEE associated with the address space is used to locate the user ID.

Format

```
CALL IRRSPX00 (Work_area,
              ALET, SAF_return_code,
              ALET, RACF_return_code,
              ALET, RACF_reason_code,
              Number_parameters,
              Function_code,
              Attributes,
              Log_string,
              Parm_list_version,
              Function_parmlist
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_Return_Code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_Return_Code

The name of a fullword in which the service routine stores the return code.

RACF_Reason_Code

The name of a fullword in which the service routine stores the reason code.

Number_parameters

Specifies the name of a fullword which contains the number of parameters that follow in the non-request specific portion of the R_PKIServ callable service invocation.

Function_code

The name of a 2-byte area containing the Function code. The function code has one of the following values:

End user functions—

- X'0001'-Generate a basic X.509V3 certificate using the application provided data pointed to by the function specific parameter list (Function name GENCERT).
- X'0002'-Extract certificate by certificate request ID (Function name EXPORT).
- X'0009'-Submit a request for a X.509V3 certificate using the application provided data pointed to by the function specific parameter list. Similar to function 1, except that the request would be pending the approval of the PKI Services administrator.(Function name REQCERT)
- X'000A'-Verify certificate was issued by this PKI Services CA and return certificate fields.(Function name VERIFY)
- X'000B'-Revoke a PKI Services certificate.(Function name REVOKE)
- X'000C'-Generate a renewal PKI Services certificate.(Function name GENRENEW)
- X'000D'-Request a renewal certificate from PKI Services.(Function name REQRENEW)

Administrative functions—

- X'0003'-Query PKI Services for certificate requests.(Function name QUERYREQS)
- X'0004'-Get detail information pertaining to one PKI Services certificate request.(Function name REQDETAILS)
- X'0005'-Modify PKI Services certificate requests.(Function name MODIFYREQS)
- X'0006'-Query PKI Services issued certificates.(Function name QUERYCERTS)
- X'0007'-Get detail information pertaining to one PKI Services issued certificate.(Function name CERTDETAILS)

- X'0008'-Modify PKI Services issued certificates.(Function name MODIFYCERTS)

Attributes

The name of a 4-byte area containing bit settings that direct the function to be performed. This is a reserved field that must be specified. The bit settings are mapped as follows:

- Functions GENCERT (X'0001) and GENRENEW (X'000c')-x'80000000' - Do not return control until the certificate has been generated. The request will be purged if unsuccessful for any reason.
All other bit positions are reserved and must be set to zero.
- All other functions
All bit positions are reserved and must be set to zero.

Log_string

The name of an area that consists of a 1-byte length field followed by character data to be included in any audit records that are created as a result of the R_PKIServ invocation. The first eight bytes of the Log_string data specified on a GENCERT and RENEW request is also used as application data(ApplData) to be stored with the certificate. If not specified, the length must equal 0.

Parmlist_version

The name of a 4-byte input value which contains the version number for the following input field, Function_parmlist. The contents of this field must be set to binary zero.

Function_parmlist

Specifies the name of the function code specific parameter list for the Function_code specified:

Table 76. Function_parmlist for GENCERT

Field	Attributes	Usage	Description
Eyecatcher	8 characters	In	Eyecatcher, 8 characters left justified blank filled. Actual value set by invoker, for example 'GENCERT'.
CertPlistLen	4-byte length	In	Describes the length in bytes of the certificate generation plist.
CertPlist	Address of	In	The name of the area which is the CertGen request parameter list. This area maps the specific name, length, address/data values which are used in satisfying the certificate request for the specified user. Also, see Table 77 on page 160.
Certid	Address of	In/Out	Points to a 57-byte area, in which the first byte will contain the actual length on return of the certificate request ID. The storage address specified must be obtained by the caller, and freed by the caller. The returned certificate request ID is used to extract the completed certificate, if the request has been accepted by RACF.

The GENCERT and REQCERT functions have in essence two connected parameter list areas; the function specific parameter list as defined above and the CertGen request parameter list (CertPlist) containing specific certificate field information. CertPlist is a list of ordered triplets which consists of name,

length, and data value. The field name is a fixed 12 character field, case sensitive, left justified, and padded with blanks, the length field is a binary four byte value, which qualifies the length of the data item. Note that all data values are EBCDIC character data unless otherwise indicated. The following tables describes the valid certificate request fields:

Table 77. CertPlist for GENCERT

Field Name	Max Length	Description
DiagInfo	80 bytes (exactly)	EyeCatcher to identify this request in virtual storage for diagnostic reasons. For certificate generation warnings and errors, RACF will update the field with diagnostic information. The length will also be updated. Required Field must be the first field in the CertPlist.
Email	64 bytes	Subject's e-mail address for distinguished name. Optional, no default. Only valid with PKI Services requests.
CommonName	64 bytes	Subject's common name. Optional, no default, except if specified with a null value (length 0), RACF will use the PGMRNAME field from the RACF user profile as determined by the UserID field for this request. If PGMRNAME is null, the common name will be of the form of RACF UserID:(user's-racf-identify), for example RACF UserID:JDOE
Title	64 bytes	Subject's Title. Optional, no default.
OrgUnit	64 bytes	Subject's Organizational Unit. Note that this field may be repeated. RACF concatenates in the order of appearance to construct the hierarchy of organizational units. Optional, no default.
Org	64 bytes	Subject's Organization. Optional, no default.
Street	64 bytes	Subject's street address. Optional, no default. Only valid with PKI Services requests.
Locality	64 bytes	Subject's City or Locality. Optional, no default.
StateProv	64 bytes	Subject's State/Providence. Optional, no default.
PostalCode	64 bytes	Subject's zip code or postal code. Optional, no default. Only valid with PKI Services requests.
Country	64 bytes	Subject's Country. Optional, no default.
KeyUsage	20 bytes	KeyUsage extension entry. One of 'handshake' 'dataencrypt' 'certsign' 'docsign' (case insensitive, do not use quotes). Note this field may be repeated to request multiple usages. Also, PKI does not allow certsign with handshake and dataencrypt. Optional, no default.
NotBefore	2 bytes	NotBefore extension entry. Number of days from today's date that the certificate becomes valid. Range is 0-30. Validity checked by RACF. Optional. Default is 0.
NotAfter	4 bytes	NotAfter extension entry. Number of days from today's date that the certificate expires. Range is 1-3650. Validity checked by RACF. Optional. Default is 365. The start of the validity period is set from the original certificate's start of validity.

Table 77. CertPlist for GENCERT (continued)

Field Name	Max Length	Description
AltIPAddr	15 bytes	Subject's Alternative Name extension entry. Dotted decimal V4 IP address for alternate name. Optional, no default.
AltURI	255 bytes	Subject's Alternative URI extension entry. Uniform Resource Identifier for alternate name. Optional, no default.
AltEmail	100 bytes	Subject's Alternative Email extension entry. E-mail address for alternate name. Optional, no default.
AltDomain	100 bytes	Subject's Alternative Domain extension entry. Domain Name for alternate name. Optional, no default.
NotifyEmail	64 bytes	E-mail address for notification purposes. Optional, no default. Only valid with PKI Services requests.
UserId	8 bytes	Subject's RACF UserID. If not specified, the userID is taken from the ACEE. For SAF requests, this is the User ID that will own the certificate. For PKI requests, the User ID is used only to determine the Common Name when CommonName is specified without a value.
Label	32 bytes	Up to 32 mixed case characters which may be used as the 'handle'. This is optional. Default is one will be generated and added to the user's list of certificates. Only valid with SAF requests.
SignWith	45 bytes	Label of z/OS Certificate authority certificate to sign the completed certificate request. The format is SAF:CERTAUTH/(ca-cert-label) or SAF:/(ca-cert-label), where ca-cert-label is the certificate label under CERTAUTH or the caller's UserID. May also be used to indicate PKI Services should process the request rather than SAF. In this case, the format is PKI:(exactly 4 characters). This is a required field.
PublicKey	65535 bytes	PKCS #10 or Netscape Navigator certificate request containing the public key to be certified. This is base 64 encoded DER. Required field.
HostIdMap	100 bytes	HostIdMapping extension entry in the form of an email address, e.g., gumby@plpsc.pok.ibm.com. The right most '@' is used to delineate the subjectId from the hostName. Optional, no default. Only valid with PKI Services requests. Note this field may be repeated.
Requestor	32 bytes	Name of the person submitting the request. Optional, derived from the first RDN of the subject's name if not specified. Only valid with PKI Services requests.
PassPhrase	32 bytes	Value to be used for challenge/response when retrieving the certificate through function EXPORT. Optional, no default. Only valid with PKI Services requests.

Table 78. Function_parmlist for EXPORT

Field	Attributes	Usage	Description
Eyecatcher	8 characters	In	Eyecatcher, 8 characters left justified blank filled. Actual value set by invoker, for example 'EXPORT'.
CertAnchorLen	4 byte length	In/Out	4-byte area which is the length of the preallocated storage of the CertAnchor area on input to EXPORT. RACF will update this value with the actual length of the certificate returned. In the event that the storage area as specified by the CertAnchor address is too small, RACF will set a failing return/reason code and update the length field to the size required. The caller must allocate a larger area.
CertAnchor	Address of	In/Out	The address of the storage area in which the R_PKIServ service stores the certificate that is specified by the CertID parameter if the service was able to successfully retrieve the completed certificate. If the caller has supplied an area which is too small, based on the CertAnchorLen, this service fails the request, and updates the CertAnchorLen field to indicate the actual storage required to store the certificate.
CertId	Address of	In	Points to a 57-byte area, in which the first byte will contain the actual length of the input certificate request ID that will be used to locate the certificate to be exported. For PKI Services requests where PassPhrase was specified on the GENCERT, the user provided pass phrase must be appended to the actual CertID value and included here. The leading length byte must account for the additional length.

Table 79. Function_parmlist for QUERYREQS

Field	Attributes	Usage	Description
Eyecatcher	8 characters	In	Eyecatcher, 8 characters left justified blank filled. Actual value set by invoker, e.g. 'QUERYRQS'
ResultsListLen	4 byte length	In/Out	4 byte area which is the length of the pre-allocated storage of the Results List area on input to QUERYREQS. RACF will update this value with the actual length of the data returned. In the event that the storage area as specified by the Results List address is too small, RACF will set a failing return/reason code and update the length field to the size required. The caller must allocate a larger area.

Table 79. Function_parmlist for QUERYREQS (continued)

Field	Attributes	Usage	Description
ResultsList	Address of	In/Out	The address of the storage area in which the R_PKIServ service stores the results of the query if the service was able to successfully retrieve the data. If the caller has supplied an area which is too small, (based in the ResultsListLen), this service fails the request, and updates the ResultsListLen field to indicate the actual storage required to store the data. Also, see Table 80 on page 164.
Certid	Address of	In	Points to a 57 byte area, in which the first byte will contain the actual length of the input certificate request ID that will be used as a starting point for this query. Only requests located after this request will be returned. If the first byte is zero (x'00'), the query will start with the first request.
NumEntries	4 byte numeric	In/Out	Input value indicating the maximum number of entries that should be returned in the ResultsList area. Zero indicates no limit. Updated to indicate the number of entries actually returned.
CriteriaStatus	4 byte numeric	In	Value indicating the request status to use as search criteria. X'00000000' - return all requests, X'00000001' - return requests pending approval only, X'00000002' - return requests that have been approved only, X'00000003' - return completed requests only, X'00000004 - return all rejected requests only, X00000005 - return rejected requests in which the client has been notified only.
CriteriaDays	4 byte numeric	In	Value indicating the recent activity time period to use as additional search criteria. The time period is the number of days in the past that should be scanned for requests that have been created or modified. If zero (x'00000000'), recent activity will not be used as additional search criteria.
CriteriaName	Address of	In	Points to a 33 byte area, in which the first byte will contain the actual length of the input requestor's name to be used as additional search criteria. If the first byte is zero (x'00'), the requestor's name will not be used as additional search criteria.

The QUERYREQS function returns results in the ResultsList area provided by the caller. The ResultsList has the following format:

Table 80. ResultsList for QUERYREQS

1 byte length	Entry 1's certificate request ID, max 56 bytes
1 byte length	Entry 1's requestor's name, max 32 bytes
1 byte length	Entry 1's subject's distinguished name, max 255 bytes
1 byte length	Entry 1's issuer's distinguished name, max 255 bytes
1 byte length	Entry 1's validity period in local time. Format YYYY/MM/DD HH:MM:SS - YYYY/MM/DD HH:MM:SS. Exactly 41 bytes
1 byte length	Entry 1's keyUsage value, max 64 bytes (one or more of 'handshake', 'dataencrypt', 'certsign', 'docsign', or "not specified")
1 byte length	Entry 1's status, max 32 bytes (one of "Pending Approval", "Approved", "Completed", "Rejected", or "Rejected, User Notified")
1 byte length	Entry 1's creation date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry 1's last modified date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry 1's ApplData value from the GENCERT or REQCERT invocation, max 8 bytes
1 byte length	Entry 1's serial number if certificate has been issued, max 16 bytes
1 byte length	Entry 1's previous serial number if this is a renewal request, max 16 bytes
1 byte length	Entry 2's certificate request ID, max 56 bytes
1 byte length	Entry 2's requestor's name, max 32 bytes
1 byte length	Entry 2's subject's distinguished name, max 255 bytes
1 byte length	Entry 2's issuer's distinguished name, max 255 bytes
1 byte length	Entry 2's validity period in local time. Format YYYY/MM/DD HH:MM:SS - YYYY/MM/DD HH:MM:SS. Exactly 41 bytes
1 byte length	Entry 2's keyUsage value, max 64 bytes (one or more of 'handshake', 'dataencrypt', 'certsign', 'docsign', or "not specified")
1 byte length	Entry 2's status, max 32 bytes (one of "Pending Approval", "Approved", "Completed", "Rejected", or "Rejected, User Notified")
1 byte length	Entry 2's creation date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry 2's last modified date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry 2's ApplData value from the GENCERT or REQCERT invocation, max 8 bytes
1 byte length	Entry 2's serial number if certificate has been issued, max 16 bytes
1 byte length	Entry 2's previous serial number if this is a renewal request, max 16 bytes

⋮

1 byte length	Entry n's certificate request ID, max 56 bytes
1 byte length	Entry n's requestor's name, max 32 bytes
1 byte length	Entry n's subject's distinguished name, max 255 bytes
1 byte length	Entry n's issuer's distinguished name, max 255 bytes
1 byte length	Entry n's validity period in local time. Format YYYY/MM/DD HH:MM:SS - YYYY/MM/DD HH:MM:SS. Exactly 41 bytes
1 byte length	Entry n's keyUsage value, max 64 bytes (one or more of 'handshake', 'dataencrypt', 'certsign', 'docsign', or "not specified")
1 byte length	Entry n's status, max 32 bytes (one of "Pending Approval", "Approved", "Completed", "Rejected", or "Rejected, User Notified")
1 byte length	Entry n's creation date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry n's last modified date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry n's ApplData value from the GENCERT or REQCERT n's last modified date in YYYY/MM/DD invocation, max 8 bytes
1 byte length	Entry n's serial number if certificate has been issued, max 16 bytes
1 byte length	Entry n's previous serial number if this is a renewal request, max 16 bytes

Table 81. Function_parmlist for REQDETAILS

FIELD	ATTRIBUTES	USAGE	DESCRIPTION
Eyecatcher	8 characters	In	Eyecatcher, 8 characters left justified blank filled. Actual value set by invoker, e.g. 'REQDTAIL'
SumListLen	4 byte length	In/Out	4 byte area which is the length of the pre-allocated storage of the Summary List area on input to REQDETAILS. RACF will update this value with the actual length of the data returned. In the event that the storage area as specified by the Summary List address is too small, RACF will set a failing return/reason code and update the length field to the size required. The caller must allocate a larger area.
SumList	Address of	In/Out	The address of the storage area in which the R_PKIServ service stores the results of the query if the service was able to successfully retrieve the data. If the caller has supplied an area which is too small, (based in the SummaryListLen), this service fails the request, and updates the SummaryListLen field to indicate the actual storage required to store the data. Also, see Table 82 on page 166.

Table 81. Function_parmlist for REQDETAILS (continued)

FIELD	ATTRIBUTES	USAGE	DESCRIPTION
CertPlistLen	4 byte length	In/Out	4 byte area which is the length of the pre-allocated storage of the certificate generation plist area on input to REQDETAILS. RACF will update this value with the actual length of the data returned. In the event that the storage area as specified by the Results List address is too small, RACF will set a failing return/reason code and update the length field to the size required. The caller must allocate a larger area.
CertPlist	Address of	In/Out	The address of the storage area in which the R_PKIServ service stores the results of the query if the service was able to successfully retrieve the data. If the caller has supplied an area which is too small, (based in the CertPlistLen), this service fails the request, and updates the CertPlistLen field to indicate the actual storage required to store the data.. This area maps some of the specific name, length, address/data values which were used when generating the original certificate request on GENCERT. Also, see Table 83 on page 167.
CertId	Address of	In	Points to a 57 byte area, in which the first byte will contain the actual length of the input certificate request ID from which details are to be extracted

The REQDETAILS function returns QUERYREQS style summary data in the SumList area provided by the caller. The SumList has the following format:

Table 82. SumList for REQDETAILS

1 byte length	Entry's certificate request ID, max 56 bytes
1 byte length	Entry's requestor's name, max 32 bytes
1 byte length	Entry's subject's distinguished name, max 255 bytes
1 byte length	Entry's issuer's distinguished name, max 255 bytes
1 byte length	Entry's validity period in local time. Format YYYY/MM/DD HH:MM:SS - YYYY/MM/DD HH:MM:SS. Exactly 41 bytes
1 byte length	Entry's keyUsage value, max 64 bytes (one or more of 'handshake', 'dataencrypt', 'certsign', 'docsign', or "not specified")
1 byte length	Entry's status, max 32 bytes (one of "Pending Approval", "Approved", "Completed", "Rejected", or "Rejected, User Notified")
1 byte length	Entry's creation date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry's last modified date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry's ApplData value from the GENCERT or REQCERT invocation, max 8 bytes
1 byte length	Entry's serial number if certificate has been issued, max 16 bytes

Table 82. SumList for REQDETAILS (continued)

1 byte length	Entry's previous serial number if this is a renewal request, max 16 bytes
1 byte length	Entry's last action comment, max 64 bytes
1 byte length	Entry's pass phrase provided when the certificate request was made, max 32 bytes
1 byte length	Entry's notification e-mail address, max 64 bytes

Additionally, the REQDETAILS function returns GENCERT style certificate field name/value pairs in the CertPlist area. This is the list of fields that may be returned. They are also the fields that may be modified by function MODIFYREQS. The fields and their values are conditionally present, depending on the values of the original GENCERT request. Multiple OrgUnits and HostIdMaps are returned in the order they were originally specified. Fields other than OrgUnit and HostIdMap are not returned in any specific order. Like GENCERT, the CertPlist returned is a list of ordered triplets which consists of name, length and data value. The field name is a fixed 12 character field, case sensitive, left justified, and padded with blanks, the length field is a binary four byte value, which qualifies the length of the data item.

Note: All data values are EBCDIC character data unless otherwise indicated.

Also, NotBefore and NotAfter are replaced with StartDate and EndDate.

Table 83. CertPlist for REQDETAILS

Field Name	MaxLength	Description
Email	64 bytes	Subject's e-mail address for distinguished name
CommonName	64 bytes	Subject's common name
Title	64 bytes	Subject's title
OrgUnit	64 bytes	Subject's Organization Unit. Note this field may be repeated
Org	64 bytes	Subject's Organization
Street	64 bytes	Subject's street address
Locality	64 bytes	Subject's City or Locality
StateProv	64 bytes	Subject's State or Province
PostalCode	64 bytes	Subject's zip code or postal code
Country	2 bytes	Subject's Country
KeyUsage	20 bytes	KeyUsage extension entry. One of 'handshake' 'dataencrypt' 'certsign' 'docsign' (case insensitive, no quotes). Note this field may be repeated.
StartDate	10 bytes	Data certificate becomes valid in YYYY/MM/DD form.
EndDate	10 bytes	Last date that the certificate is valid in YYYY/MM/DD form.
AltIPAddr	15 bytes	Subject's Alternative Name extension entry. Dotted decimal V4 IP address for alternate name.

Table 83. CertPlist for REQDETAILS (continued)

Field Name	MaxLength	Description
AltURI	255 bytes	Subject's Alternative URI extension entry. Uniform Resource Identifier for alternate name.
AltEmail	100 bytes	Subject's Alternative Email extension entry. E-mail address for alternate name.
AltDomain	100 bytes	Subject's Alternative Domain extension entry. Domain Name for alternate name.
HostIdMap	100 bytes	HostIdMapping extension entry. Note this field may be repeated.

Table 84. Function_parmlist for MODIFYREQS

Field	Attributes	Usage	Description
Eyecatcher	8 characters	In	Eyecatcher, 8 characters left justified blank filled. Actual value set by invoker, e.g. 'MODREQS'
Action	4 byte value	In	4 byte binary value indicating the action to take against the requests. X'00000001' - Approve with possible modifications as specified below, X'00000002' - Reject, X'00000003' -Delete from request database.
Comment	Address of	In	Points to a 65 byte area, in which the first byte will contain the actual length of the comment associated with this action. If the first byte is zero (x'00'), no comment will be recorded.
CertIdsLen	4 byte length	In/Out	Describes the length in bytes of the input certificate request ID list. May be modified by RACF on output if a smaller list is being returned.
CertIds	Address of	In/Out	Points to an area containing 1 or more certificate request Ids that are to be modified by this request. Each certificate request ID occupies a maximum of 57 bytes, in which the first byte will contain the actual length of the certificate request ID. If any requests cannot be modified because their states changed,RACF will return a shortened list containing those request Ids that couldn't be modified .
CertPlistLen	4 byte length	In	Describes the length in bytes of the certificate modification plist. A zero indicates no modification plist.

Table 84. Function_parmlist for MODIFYREQS (continued)

Field	Attributes	Usage	Description
CertPlist	Address of	In	Points to the area which is the certificate modification parameter list. This area maps the specific name, length, address/data values which are used to replace the existing values in the certificate request. The format is the same as the Certificate Request Plist defined under GENCERT (DiagInfo must be the first field.), except that the modifiable fields are those listed below. The certificate modification plist is valid for the "Approve" action only and only when the CertIds list contains exactly one Certificate ID. For all other cases, it is ignored. If no modification plist is specified for "Approve", the request is approved as is. If a modification plist is specified, the existing request values for the key usage, start and end validity, alternate names, and hostid mappings are all deleted. The request is then repopulated with the values specified in the modification plist. Any desired value must be specified in the modification plist, even if it is not changing. The subject's distinguished name fields work slightly differently. If no subject's distinguished name fields are specified in the modification plist, the existing values are retained. If any subject's distinguished name fields are specified, the name is completely replaced by the values specified. Also, see Table 85.

The MODIFYREQS modification plist (CertPlist) Structure. Like GENCERT, the CertPlist is a list of ordered triplets which consists of name, length and data value. The field name is a fixed 12 character field, case sensitive, left justified, and padded with blanks, the length field is a binary four byte value, which qualifies the length of the data item. Note, all data values are EBCDIC character data unless otherwise indicated. Note that NotBefore and NotAfter are replaced with StartDate and EndDate. See GENCERT for more information on the other individual fields

Table 85. CertPlist for MODIFYREQS

Field Name	Max Length	Description
DiagInfo	80 bytes (exactly)	Diagnostic information area. Must be first field in the CertPlist. For certificate generation warnings and errors, RACF will update this field with diagnostic information. The length will be updated as well. Required field.
Email	64 bytes	Subject's e-mail address for distinguished name. Optional.
CommonName	64 bytes	Subject's common name. Optional.
Title	64 bytes	Subject's title. Optional.
OrgUnit	64 bytes	Subject's Organizational Unit. Note this field may be repeated. Optional.
Org	64 bytes	Subject's Organization. Optional.

Table 85. CertPlist for MODIFYREQS (continued)

Field Name	Max Length	Description
Street	64 bytes	Subject's street address. Optional.
Locality	64 bytes	Subject's City or Locality. Optional.
StateProv	64 bytes	Subject's State or Province. Optional.
PostalCode	64 bytes	Subject's Zip or postal code. Optional.
Country	2 bytes	Subject's Country. Optional.
KeyUsage	20 bytes	KeyUsage extension entry. One of 'handshake' 'dataencrypt' 'certsign' 'docsign' (case insensitive, no quotes). Note this field may be repeated. Optional.
StartDate	10 bytes	Date certificate becomes valid in YYYY/MM/DD form. Must be a valid date within the range 1970/01/01 through 2036/12/31. Required.
EndDate	10 bytes	Last date that the certificate is valid in YYYY/MM/DD form. Must be a valid date within the range of today through 2036/12/31 and must not be prior to StartDate. Required.
AltIPAddr	15 bytes	Subject's Alternative Name extension entry. Dotted decimal V4 IP address for alternate name. Optional.
AltURL	255 bytes	Subject's Alternative URI extension entry. Uniform Resource Identifier for alternate name. Optional.
AltEmail	100 bytes	Subject's Alternative Email extension entry. E-mail address for alternate name. Optional.
AltDomain	100 bytes	Subject's Alternative Domain extension entry. Domain Name for alternate name. Optional.
HostIdMap	100 bytes	HostIdMapping extension entry. Note this field may be repeated. Optional.

Table 86. Function_parmlist for QUERYCERTS

Field	Attributes	Usage	Description
Eyecatcher	8 characters	In	Eyecatcher, 8 characters left justified blank filled. Actual value set by invoker, e.g. ' QUERYCTS '
ResultsListLen	4 byte length	In/Out	4 byte area which is the length of the pre-allocated storage of the Results List area on input to QUERYCERTS. RACF will update this value with the actual length of the data returned. In the event that the storage area as specified by the Results List address is too small, RACF will set a failing return/reason code and update the length field to the size required. The caller must allocate a larger area.

Table 86. Function_parmlist for QUERYCERTS (continued)

Field	Attributes	Usage	Description
ResultsList	Address of	In/Out	The address of the storage area in which the R_PKIServ service stores the results of the query if the service was able to successfully retrieve the data. If the caller has supplied an area which is too small, (based in the ResultsListLen), this service fails the request, and updates the ResultsListLen field to indicate the actual storage required to store the data. Also, see Table 87.
SerialNum	Address of	In	Points to a 17 byte area, in which the first byte will contain the actual length of the input certificate serial number that will be used as a starting point for this query. Only certificates located after this certificate will be returned. If the first byte is zero (x'00'), the query will start with the first request. The serial number is in printable EBCDIC (HEX) form e.g., "01A6",
NumEntries	4 byte numeric	In/Out	Input value indicating the maximum number of entries that should be returned in the ResultsList area. Zero indicates no limit. Updated to indicate the number of entries actually returned.
CriteriaStatus	4 byte numeric	In	Value indicating the certificate status to use as search criteria. X'00000000' - return all issued certificates, X'00000001' - return revoked certificates only, X'00000002' -return expired certificates only, X'00000003' - return non-expired, non-revoked certificates only (i.e., active certificates), X'00000004' - return non-expired revoked certificates only (i.e., CRL certificates).
CriteriaDays	4 byte numeric	In	Value indicating the recent activity time period to use as additional search criteria. The time period is the number of days in the past that should be scanned for certificates that have been created or modified. If zero (x'00000000'), recent activity will not be used as additional search criteria .
CriteriaName	Address of	In	Points to a 33 byte area, in which the first byte will contain the actual length of the input requestor's name to be used as additional search criteria. If the first byte is zero (x'00'), the requestor's name will not be used as additional search

The QUERYCERTS function returns results in the ResultsList area provided by the caller. The ResultsList has the following format:

Table 87. ResultsList for QUERYCERTS

1 byte length	Entry 1's serial number in printable EBCDIC (HEX) form e.g., "01A6", max 16 bytes
1 byte length	Entry 1's requestor's name, max 32 bytes

Table 87. ResultsList for QUERYCERTS (continued)

1 byte length	Entry 1's subject's distinguished name, max 255 bytes
1 byte length	Entry 1's issuer's distinguished name, max 255 bytes
1 byte length	Entry 1's validity period in local time. Format YYYY/MM/DD HH:MM:SS - YYYY/MM/DD HH:MM:SS. Exactly 41 bytes
1 byte length	Entry 1's keyUsage value, max 64 bytes (one or more of 'handshake', 'dataencrypt', 'certsign', 'docsign', or "not specified")
1 byte length	Entry 1's status, max 32 bytes, one of "Active", "Expired", "Revoked", or "Revoked, Expired"
1 byte length	Entry 1's creation date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry 1's last modified date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry 1's ApplData value from the GENCERT or REQCERT invocation, max 8 bytes
1 byte length	Entry 2's serial number in printable EBCDIC (HEX) form e.g., "01A6", max 16 bytes
1 byte length	Entry 2's requestor's name, max 32 bytes
1 byte length	Entry 2's subject's distinguished name, max 255 bytes
1 byte length	Entry 2's issuer's distinguished name, max 255 bytes
1 byte length	Entry 2's validity period in local time. Format YYYY/MM/DD HH:MM:SS - YYYY/MM/DD HH:MM:SS. Exactly 41 bytes
1 byte length	Entry 2's keyUsage value, max 64 bytes (one or more of 'handshake', 'dataencrypt', 'certsign', 'docsign', or "not specified")
1 byte length	Entry 2's status, max 32 bytes, one of "Active", "Expired", "Revoked", or "Revoked,Expired"
1 byte length	Entry 2's creation date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry 2's last modified date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry 2's ApplData value from the GENCERT or REQCERT invocation, max 8 bytes

⋮

1 byte length	Entry n's serial number in printable EBCDIC(HEX) form e.g., "01A6", max 16 bytes
1 byte length	Entry n's requestor's name, max 32 bytes
1 byte length	Entry n's subject's distinguished name, max 255 bytes
1 byte length	Entry n's issuer's distinguished name, max 255 bytes
1 byte length	Entry n's validity period in local time. Format YYYY/MM/DD HH:MM:SS - YYYY/MM/DD HH:MM:SS. Exactly 41 bytes
1 byte length	Entry n's keyUsage value, max 64 bytes (one or more of 'handshake', 'dataencrypt', 'certsign', 'docsign', or "not specified")
1 byte length	Entry n's status, max 32 bytes, one of "Active", "Expired", "Revoked", "Revoked,Expired"

1 byte length	Entry n's creation date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry n's last modified date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry n's ApplData value from the GENCERT or REQCERT invocation, max 8 bytes

Table 88. Function_parmlist for CERTDETAILS

Field	Attributes	Usage	Description
Eyecatcher	8 characters	In	Eyecatcher, 8 characters left justified blank filled. Actual value set by invoker, e.g. 'CRTDTAIL'
SumListLen	4 byte length	In/Out	4 byte area which is the length of the pre-allocated storage of the Summary List area on input to CERTDETAILS. RACF will update this value with the actual length of the data returned. In the event that the storage area as specified by the Summary List address is too small, RACF will set a failing return/reason code and update the length field to the size required. The caller must allocate a larger area.
SumList	Address of	In/Out	The address of the storage area in which the R_PKIServ service stores the results of the query if the service was able to successfully retrieve the data. If the caller has supplied an area which is too small, (based in the SummaryListLen), this service fails the request, and updates the SummaryListLen field to indicate the actual storage required to store the data. Also, see Table 89 on page 174.
CertPlistLen	4 byte length	In/Out	4 byte area which is the length of the pre-allocated storage of the certificate generation plist area on input to CERTDETAILS. RACF will update this value with the actual length of the data returned. In the event that the storage area as specified by the Results List address is too small, RACF will set a failing return/reason code and update the length field to the size required. The caller must allocate a larger area.
CertPlist	Address of	In/Out	The address of the storage area in which the R_PKIServ service stores the results of the query if the service was able to successfully retrieve the data. If the caller has supplied an area which is too small, (based in the CertPlistLen), this service fails the request, and updates the CertPlistLen field to indicate the actual storage required to store the data.. This area maps some of the specific name, length, address/data values which were used when generating the original certificate request on GENCERT. Also, see Table 90 on page 174.

Table 88. Function_parmlist for CERTDETAILS (continued)

Field	Attributes	Usage	Description
SerialNum	Address of	In	Points to a 17 byte area, in which the first byte will contain the actual length of the input certificate serial number from which details are to be extracted. The serial number is in printable EBCDIC (HEX) form e.g., "01A6",

The CERTDETAILS function returns QUERYCERTS style summary data in the SumList area provided by the caller. The SumList has the following format:

Table 89. SumList for CERTDETAILS

1 byte length	Entry's serial number in printable EBCDIC(HEX) form e.g., "01A6", max 16 bytes
1 byte length	Entry's requestor's name, max 32 bytes
1 byte length	Entry's subject's distinguished name, max 255 bytes
1 byte length	Entry's issuer's distinguished name, max 255 bytes
1 byte length	Entry's validity period in local time. Format YYYY/MM/DD HH:MM:SS - YYYY/MM/DD HH:MM:SS. Exactly 41 bytes
1 byte length	Entry's keyUsage value, max 64 bytes (one or more of 'handshake', 'dataencrypt', 'certsign', 'docsign', or "not specified")
1 byte length	Entry's status, max 32 bytes, one of "Active", "Expired", "Revoked", "Revoked,Expired"
1 byte length	Entry's creation date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry's last modified date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry's ApplData value from the GENCERT or REQCERT invocation, max 8 bytes
1 byte length	Entry's last action comment, max 64 bytes

Additionally, the CERTDETAILS function returns GENCERT style certificate field name/value pairs in the CertPlist area. This is the list of fields that may be returned. The fields and their values are conditionally present, depending on the values of the original GENCERT request. Multiple HostIdMaps are returned in the order they were originally specified. Fields other than HostIdMap are not returned in any specific order. Like GENCERT, the CertPlist returned is a list of ordered triplets which consists of name, length and data value. The field name is a fixed 12 character field, case sensitive, left justified, and padded with blanks, the length field is a binary four byte value, which qualifies the length of the data item. Note, all data values are EBCDIC character data unless otherwise indicated.

Table 90. CertPlist for CERTDETAILS

Field	Max Length	Description
AltIPAddr	15 bytes	Subject's Alternative Name extension entry. Dotted decimal V4 IP address for alternate name. Optional.
AltURI	255 bytes	Subject's Alternative URI extension entry. Uniform Resource Identifier for alternate name. Optional.

Table 90. CertPlist for CERTDETAILS (continued)

Field	Max Length	Description
AltEmail	100 bytes	Subject's Alternative Email extension entry. E-mail address for alternate name. Optional.
AltDomain	100 bytes	Subject's Alternative Domain extension entry. Domain Name for alternate name. Optional.
HostIdMap	100 bytes	HostIdMapping extension entry. Note this field may be repeated. Optional.

Table 91. Function_parmlist for MODIFYCERTS

Field	Attributes	Usage	Description
Eyecatcher	8 characters	In	Eyecatcher, 8 characters left justified blank filled. Actual value set by invoker, e.g. 'MODCERTS'
Action	4 byte value	In	4 byte binary value indicating the action to take against the certificates. X'00000002' - Revoke, X'00000003' - Delete from issued certificate database.
Comment	Address of	In	Points to a 65 byte area, in which the first byte will contain the actual length of the comment associated with this action. If the first byte is zero (x'00'), no comment will be recorded.
SerialNumsLen	4 byte length	In/Out	Describes the length in bytes of the input certificate serial number list. May be modified by RACF on output if a smaller list is being returned.
SerialNums	Address of	In/Out	Points to an area containing 1 or more certificate serial numbers that are to be modified by this request. Each certificate request ID occupies a maximum of 17 bytes, in which the first byte will contain the actual length of the certificate serial number. The serial number itself is in printable EBCDIC (HEX) form e.g., "01A6". If any certificates cannot be modified because their states changed, RACF will return a shortened list containing those serial numbers that couldn't be modified .
Reason	4 byte value	In	4 byte binary value indicating the reason for the certificate revocation, X'00000000' - No Reason, X'00000001' - User key was compromised, X'00000002' - CA key was compromised, X'00000003' - User changed affiliation, X'00000004' - Certificate was superseded, X'00000005' -Original use no longer valid. Ignored for Actions other than "Revoke"

Table 92. Function_parmlist for VERIFY

Field	Attributes	Usage	Description
Eyecatcher	8 characters	In	Eyecatcher, 8 characters left justified blank filled. Actual value set by invoker, e.g. 'VERIFY'.

Table 92. Function_parmlist for VERIFY (continued)

Field	Attributes	Usage	Description
SumListLen	4 byte length	In/Out	4 byte area which is the length of the pre-allocated storage of the Summary List area on input to VERIFY. RACF will update this value with the actual length of the data returned. In the event that the storage area as specified by the Summary List address is too small, RACF will set a failing return/reason code and update the length field to the size required. The caller must allocate a larger area.
SumList	Address of	In/Out	The address of the storage area in which the R_PKIServ service stores the results of the verify if the service was able to successfully retrieve the data. If the caller has supplied an area which is too small, (based in the SummaryListLen) this service fails the request, and updates the SummaryListLen field to indicate the actual storage required to store the data. Also, see Table 93.
CertPlistLen	4 byte length	In/Out	4 byte area which is the length of the pre-allocated storage of the certificate generation plist area on input to VERIFY. RACF will update this value with the actual length of the data returned. In the event that the storage area as specified by the Results List address is too small, RACF will set a failing return/reason code and update the length field to the size required. The caller must allocate a larger area.
CertPlist	Address of	In/Out	The address of the storage area in which the R_PKIServ service stores the results of the verify if the service was able to successfully retrieve the data. If the caller has supplied an area which is too small, (based in the CertPlistLen) this service fails the request, and updates the CertPlistLen field to indicate the actual storage required to store the data.. This area maps some of the specific name, length, address/data values which were used when generating the original certificate request on GENCERT. Also, see Table 94 on page 177.
CertLen	4 byte length	In	4 byte area which is the length of the certificate contained in the Cert area on input to VERIFY.
Cert	Address of	In	The address of the storage area containing the certificate to verify. This is base64 encoded DER.

The VERIFY function returns CERTDETAILS style summary data in the SumList area provided by the caller. The SumList has the following format:

Table 93. SumList for VERIFY

1 byte length	Entry's serial number in printable EBCDIC(HEX) form e.g., "01A6", max 16 bytes
---------------	--

Table 93. SumList for VERIFY (continued)

1 byte length	Entry's requestor's name, max 32 bytes
1 byte length	Entry's subject's distinguished name, max 255 bytes
1 byte length	Entry's issuer's distinguished name, max 255 bytes
1 byte length	Entry's validity period in local time. Format YYYY/MM/DD HH:MM:SS - YYYY/MM/DD HH:MM:SS. Exactly 41 bytes
1 byte length	Entry's keyUsage value, max 64 bytes (one or more of 'handshake', 'dataencrypt', 'certsign', 'docsign', or "not specified")
1 byte length	Entry's status, max 32 bytes, one of "Active", "Expired", "Revoked", "Revoked,Expired"
1 byte length	Entry's creation date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry's last modified date in YYYY/MM/DD form, exactly 10 bytes
1 byte length	Entry's ApplData value from the GENCERT or REQCERT invocation, max 8 bytes

Additionally, the VERIFY function returns GENCERT style certificate field name/value pairs in the CertPlist area. The fields and their values are conditionally present, depending on the values actually contained in the certificate. Fields are not returned in any specific order. Like GENCERT, the CertPlist is a list of ordered triplets which consists of name, length and data value. The field name is a fixed 12 character field, case sensitive, left justified, and padded with blanks, the length field is a binary four byte value, which qualifies the length of the data item. Note, all data values are EBCDIC character data unless otherwise indicated.

Table 94. CertPlist for VERIFY

Field	Max Length	Description
AltIPAddr	15 bytes	Dotted decimal V4 IP address.
AltURI	255 bytes	Uniform Resource Identifier.
AltEmail	100 bytes	E-mail address.
AltDomain	100 bytes	Domain Name.
HostIdMap	100 bytes	HostIdMapping extension entry. Note this field may be repeated.

Table 95. Function_parmlist for REVOKE

Field	Attributes	Usage	Description
Eyecatcher	8 characters	In	Eyecatcher, 8 characters left justified blank filled. Actual value set by invoker, e.g. ' REVOKE '
Reason	4 byte value	In	4 byte binary value indicating the reason for the certificate revocation, X'00000000' - No Reason, X'00000001' - User key was compromised, X'00000002' - CA key was compromised, X'00000003' - User changed affiliation, X'00000004' - Certificate was superseded, X'00000005' -Original use no longer valid.

Table 95. Function_parmlist for REVOKE (continued)

Field	Attributes	Usage	Description
SerialNum	Address of	In	Points to a 17 byte area, in which the first byte will contain the actual length of the input certificate serial number for the certificate that is to be revoked . The serial number is in printable EBCDIC (HEX) form e.g., "01A6",

Table 96. Function_parmlist for GENRENEW and REQRENEW

Field	Attributes	Usage	Description
Eyecatcher	8 characters	In	Eyecatcher, 8 characters left justified blank filled. Actual value set by invoker, e.g. 'RENEW '
CertPlistLen	4 byte length	In	Describes the length in bytes of the certificate generation plist. A zero indicates no modification plist
CertPlist	Address of	In	The name of the area which is the renew request parameter list. This area maps the specific name, length, address/data values which are used in satisfying the certificate request for the specified user. Also, see Table 97.
CertId	Address of	In/Out	Points to a 57 byte area, in which the first byte will contain the actual length on return of the certificate request ID. The storage address specified must be obtained by the caller, and freed by the caller. The returned certificate request ID is used to extract the completed certificate, if the request has been accepted.
SerialNum	Address of	In	Points to a 17 byte area, in which the first byte will contain the actual length of the input certificate serial number for the certificate that is to be renewed . The serial number is in printable EBCDIC (HEX) form e.g., "01A6",

Here is the layout and supported fields for the RENEW CertPlist. Since most of the certificate information from the old certificate is reused for the new, very little new information can be specified in the RENEW CertPlist. Like GENCERT, the CertPlist is a list of ordered triplets which consists of name, length and data value. The field name is a fixed 12 character field, case sensitive, left justified, and padded with blanks, the length field is a binary four byte value, which qualifies the length of the data item. Note, all data values are EBCDIC character data unless otherwise indicated. See GENCERT for more information on the other individual fields

Table 97. CertPlist for GENRENEW and REQRENEW

Field Name	Max Length	Description
DiagInfo	80 bytes (exactly)	Diagnostic information area. Must be first field in the CertPlist. For certificate generation warnings and errors, RACF will update this field with diagnostic information. The length will be updated as well. Required field.

Table 97. CertPlist for GENRENEW and REQRENEW (continued)

Field Name	Max Length	Description
PassPhrase	32 bytes	Value to be used for challenge/response when retrieving the certificate through function EXPORT. Optional, no default.
NotAfter	4 bytes	Number of days from today's date that the certificate expires. Range 1-3650. Validity checked by RACF. Optional. Default is 365. The start of the validity period is set from the original certificate 's start of validity.
NotifyEmail	64 bytes	E-mail address for notification purposes. Optional. No default.

Return and Reason Codes

R_PKIServ may return the following values in the reason and return code parameters:

Table 98. Return and Reason Codes

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	Successful completion
4	0	0	RACF not installed
8	8	4	A parameter list error has been detected. See Usage Notes for further details
8	8	8	The caller of this service has not been RACF authorized to use this callable service
8	8	12	An internal error has occurred during RACF processing of the requested function
8	8	16	Unable to establish a recovery environment
8	8	20	Function code specified is not defined
8	8	24	Parameter list version specified is not supported
8	8	28	Certificate generation provider not available
8	12	xx	Certificate generation provider internal error. Reason code is the reason code from provider

Reason and return code parameters specific to function GENCERT and REQCERT:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
8	8	40	CertPlist has an incorrect length
8	8	44	CertPlist DiagInfo field missing or has an incorrect length

R_PKIServ

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
8	8	48	Incorrect field name specified in CertPlist. The field name is either unknown or not supported by this certificate generation provider.
8	8	52	Incorrect field value specified in CertPlist
8	8	56	Required field missing from CertPlist
8	8	60	Certificate generation provider input or environment error
8	8	76	Conflicting field names specified in CertPlist.

Reason and return code parameters specific to function EXPORT:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
8	8	40	CertAnchor area missing
8	8	44	CertAnchor area too small
8	8	48	Incorrect CertID specified
8	8	52	Incorrect CertId PassPhrase specified
8	8	56	Request is still pending approval or has yet to be issued
8	8	60	Request has been rejected by the administrator

Reason and return code parameters specific to function QUERYREQS:

SAF return code	RACF return code	RACF reason code	Explanation
8	8	40	Results list area missing.
8	8	44	Results list area too small.
8	8	48	Incorrect CertId specified.
8	8	56	Incorrect status criteria specified.
8	8	60	No requests satisfy the input criteria.

Reason and return code parameters specific to function REQDETAILS:

SAF return code	RACF return code	RACF reason code	Explanation
8	8	40	Summary list or CertPlist area missing.
8	8	44	Summary list or CertPlist area too small.
8	8	48	Incorrect CertId specified.
8	8	52	Success, however name fields not returned in CertPlist.

Reason and return code parameters specific to function MODIFYREQS:

SAF return code	RACF return code	RACF reason code	Explanation
8	8	40	CertPlist has an incorrect length.
8	8	44	CertPlist DiagInfo field missing or has an incorrect length.
8	8	48	Incorrect field name specified in CertPlist.
8	8	52	Incorrect field value specified in CertPlist.
8	8	56	Required field missing from CertPlist.
8	8	60	Certificate generation input or environment error.
8	8	64	CertIds has an incorrect length or value.
8	8	68	Incorrect Action specified.
8	8	72	Reason and return code parameters specific to function MODIFY REQ: One or more requests could not be modified because of a state change. CertIds contains the certificate request Ids that could not be modified.

Reason and return code parameters specific to function QUERYCERTS:

SAF return code	RACF return code	RACF reason code	Explanation
8	8	40	Results list area missing.
8	8	44	Results list area too small.
8	8	48	Incorrect SerialNum specified.
8	8	56	Incorrect status criteria specified.
8	8	60	No certificates satisfy the input criteria.

Reason and return code parameters specific to function CERTDETAILS:

SAF return code	RACF return code	RACF reason code	Explanation
8	8	40	Summary list or CertPlist area missing.
8	8	44	Summary list or CertPlist area too small.
8	8	48	Incorrect SerialNum specified.

Reason and return code parameters specific to function MODIFYCERTS:

SAF return code	RACF return code	RACF reason code	Explanation
8	8	40	Incorrect Reason specified.
8	8	64	SerialNums has an incorrect length or value.

R_PKIServ

SAF return code	RACF return code	RACF reason code	Explanation
8	8	68	Incorrect Action specified.
8	8	72	One or more certificates could not be modified because of a state change. SerialNums contains the certificate serial numbers that could not be modified.

Reason and return code parameters specific to function VERIFY:

SAF return code	RACF return code	RACF reason code	Explanation
8	8	40	Summary list or CertPlist area missing.
8	8	44	Summary list or CertPlist area too small.
8	8	64	Incorrect certificate specified.

Reason and return code parameters specific to function REVOKE:

SAF return code	RACF return code	RACF reason code	Explanation
8	8	40	Incorrect Reason specified.
8	8	64	SerialNum has an incorrect length or value.
8	8	72	The certificate could not be revoked because of a state change.

Reason and return code parameters specific to functions GENRENEW and REQRENEW:

SAF return code	RACF return code	RACF reason code	Explanation
8	8	40	CertPlist has an incorrect length.
8	8	44	CertPlist DiagInfo field missing or has an incorrect length.
8	8	48	Incorrect field name specified in CertPlist. The field name is either unknown or not supported by this certificate generation provider.
8	8	52	Incorrect field value specified in CertPlist.
8	8	60	Certificate generation input or environment error.
8	8	64	SerialNum has an incorrect length or value.
8	8	72	The certificate could not be renewed because of a state change.
8	8	76	Conflicting field names specified in CertPlist.

Usage Notes

1. This service is intended for use by z/OS application servers, to request the fulfillment of a X.509 V.3 certificate request.
2. An audit record will be created as a result of invoking this service which will indicate the success or failure of the attempt.
3. For GENCERT, the certificate generation provider is designated by the first 4 characters of the CertPlist SignWith value, "SAF:" for SAF requests, "PKI:" for PKI Services requests. For EXPORT, the caller designates the certificate generation provider indirectly by providing the Certificate ID returned by the provider. For REQCERT, QUERYREQS, REQDETAILS, MODIFYREQS, QUERYCERTS, CERTDETAILS, VERIFY, REVOKE, GENRENEW, REQRENEW, and MODIFYCERTS, PKI Services is used exclusively. There is no SAF equivalent for these functions.
4. The CertPlist for the GENCERT REQCERT, REQDETAILS, MODIFYREQS, VERIFY, REQRENEW, GENRENEW, and CERTDETAILS functions consists of triplets which consist of field name, field length, and data. The field name is a fixed field, 12 characters in length, and the field name must be left justified, and padded with blanks. The data length is also a fixed width field of 4 bytes which contain an integer which represents the length of the following field data.
5. The R_PKIServ service requires the caller to preallocate the 57 byte storage area which will hold the certificate ID which will be returned on a successful GENCERT, REQRENEW, GENRENEW, or REQCERT. When successful RACF will update the first byte with the actual length of the CertID. The entire 57 area must be provided for EXPORT even if the actual CertID is smaller than that.
6. The R_PKIServ service requires the caller to preallocate the storage which will hold the certificate which is being extracted via the EXPORT function code. On successful certificate retrieval, RACF will update the CertAnchorLen field with the actual length of the certificate. If the storage area is too small to hold the certificate, RACF will fail the request and update the CertAnchorLen field in the EXPORT request specific parameter list as supplied by the caller of this service. The caller is responsible for releasing and obtaining a new area of virtual storage which is the size as specified by RACF, and retrying the EXPORT operation.

Note: The retry may have to be performed more than once.

7. The R_PKIServ service requires the caller to preallocate the storage which will hold the results list which is being retrieved via the QUERYREQS and QUERYCERTS function codes. On success, RACF will update the ResultsListLen field with the actual length of the data returned. If the storage area is too small to hold the data, RACF will fail the request and update the ResultsListLen field in the request specific parameter list as supplied by the caller of this service. The caller is responsible for releasing and obtaining a new area of virtual storage which is the size as specified by RACF, and retrying the operation.
8. The R_PKIServ service requires the caller to preallocate the storage which will hold the summary list which is being retrieved via the VERIFY, REQDETAILS, and CERTDETAILS function codes. On success, RACF will update the SumListLen field with the actual length of the data returned. If the storage area is too small to hold the data, RACF will fail the request and update the SumListLen field in the request specific parameter list as supplied by the

R_PKIServ

caller of this service. The caller is responsible for releasing and obtaining a new area of virtual storage which is the size as specified by RACF, and retrying the operation.

9. The R_PKIServ service requires the caller to preallocate the storage which will hold the CertPlist which is being retrieved via the VERIFY, REQDETAILS, and CERTDETAILS function codes. On success, RACF will update the CertPlistLen field with the actual length of the data returned. If the storage area is too small to hold the data, RACF will fail the request and update the CertPlistLen field in the request specific parameter list as supplied by the caller of this service. The caller is responsible for releasing and obtaining a new area of virtual storage which is the size as specified by RACF, and retrying the operation
10. The actual values for the eyecatchers in the function specific parameters lists and the DiagInfo field in the CertPlist for GENCERT, MODIFYREQS, GENRENEW, REQRENEW, and REQCERT invocations are determined by the caller.
11. For GENCERT, REQCERT, GENRENEW, REQRENEW, and MODIFYREQS CertPlist field errors (reason codes 48, 52, 56, and 76), RACF will update the DiagInfo field with the name of the field in error. The length will also be updated.
12. For MODIFYREQS and MODIFYCERTS state change errors (reason code 72), RACF will update the CertIds or SerialNums field with the list of Certificate IDs or Serial Numbers that could not be updated. The length field will also be updated to reflect the size of the data being returned.
13. For GENCERT and REQCERT PKI Services requests, the special user Ids that start with lowercase 'irr' may not be specified for the CertPlist field UserId.
14. For GENCERT, REQCERT, GENRENEW, REQRENEW, and MODIFYREQS certificate generation errors (reason code 60) RACF will update the DiagInfo field with a product specific diagnostic message. For SAF requests, the message will have the following format: error-description (message-ID), where message-ID is the RACDCERT error message ID that is closely related to this error:

*No matching certificate was found for "SignWith" (IRRD107I)
"PublicKey" encoding does not have a valid signature (IRRD112I)
"PublicKey" encoding is not valid (IRRD104I)
"PublicKey" encoding contains an unsupported encryption algorithm (IRRD118I)
"PublicKey" extension not permitted for CERTAUTH certificate (IRRD126I)
"Label" specified is already in use (IRRD111I)
"SignWith" requires a certificate with an associated private key (IRRD128I)
Certificate cannot be added. Serial number for this CA already in use (IRRD109I)
SignWith key is an ICSF key. ICSF is not operational (IRRD135I)
Subject's name exceeds the maximum allowed characters, which is 255 (IRRD131I)*

Additionally, for successful certificate generation (reason code 0), RACF may also update the DiagInfo field with one of the following informational diagnostic messages:

Inconsistency detected. Signing certificate is not trusted (IRRD132I)
Inconsistency detected. Signing certificate's date range is incorrect (IRRD113I)

RACF may also issue its own diagnostic messages when acting as an RA for PKI Services.

For further information see *z/OS Security Server RACF Command Language Reference, SC28-1919-07* and *z/OS Security Server RACF Messages and Codes, SC28-1918-07*. It is expected that other security products which may be installed in place of RACF would have their own product specific diagnostic data.

15. For GENCERT, REQCERT, and MODIFYREQS, RACF forms the subject's distinguished name in the following order:

- Email
- CommonName
- Title
- OrgUnits (in the order that they appear in the CertPlist)
- Org
- Street
- Locality
- StateProv
- PostalCode
- Country

Except as noted below, only those portions of the name specified in the CertPlist will appear in the certificate. For GENCERT and REQCERT, if no name fields are specified in the CertPlist, the name is taken directly from the PublicKey field.

16. Different certificate generation providers may produce certificates with different extension values. To determine what certificate extensions will be created by a given provider, see the provider's supporting documentation, *z/OS Security Server RACF Security Administrator's Guide*, and *z/OS Security Server RACF Command Language Reference* or *z/OS Security Server PKI Services Guide and Reference*.
17. For GENCERT and REQCERT, if CommonName is specified with a null value (length 0), RACF will use the PGMRNAME field from the RACF user profile as determined by the UserID field for this request. If PGMRNAME is null, the common name will be of the form of RACF User ID:<*user's-racf-identity*>, for example RACF UserID:JOHNDOE. The above formula is also used for SAF requests if none of the subject's distinguished name fields are specified (Email, CommonName, Title, OrgUnit, Org, Street, Locality, StateProv, PostalCode, or Country) and the PublicKey field contains no information.
18. For GENCERT, REQCERT, GENRENEW, REQRENEW, and MODIFYREQS, all CertPlist fields specified must have a non-zero length except for CommonName, which may be null for GENCERT and REQCERT only.
19. For GENCERT, REQCERT, and MODIFYREQS, OrgUnit, HostIdMap, and KeyUsage may be repeated. For all other CertPlist fields if multiple occurrences are found, the last one will be used.
20. For GENCERT and REQCERT, the PublicKey must be either a Netscape Navigator key, a Microsoft Internet Explorer key, or a true PKCS#10 certificate request.
21. For successful EXPORTs, the certificate returned in the CertAnchor area is either a base64 encoded DER X509 certificate or a base64 encoded DER PKCS#7 certificate chain. In either case, the base64 data is wrapped with the

R_PKIServ

standard "-----BEGIN CERTIFICATE-----" header and "-----END CERTIFICATE-----" footer. For RACF requests, the returned certificate is always X.509. For PKI Services requests, the returned certificate will be packaged as a PKCS#7 certificate chain if at least one heirarchy certificate can be located under the CERTAUTH category and either subsequent access checking is not being performed or the access check user ID has CONTROL authority to IRR.DIGTCERT.EXPORT in the FACILITY Class or is RACF SPECIAL. Otherwise, an X.509 certificate is returned.

22. For GENCERT, REQCERT, GENRENEW, REQRENEW, and MODIFYREQS no validity checking is done for the following fields: AltEmail, AltDomain, AltURI. Additionally, AltPAddr, Email, NotifyEmail, and HostIdMap are checked form only (AltPAddr must be in dotted decimal form as per IP Version 4. Email, NotifyEmail, and HostIdMap must be in <subject-id>@<host-name>form).
23. For MODIFYREQS, if a modification plist (CertPlist) is specified, all the existing request values for the validity period, KeyUsage, alternate names, and HostIdMappings are completely replaced by the new values. Thus if the intent is to alter just one field, the unchanged fields must also be provided. The subject's distinguished name fields work slightly differently. If no subject's distinguished name fields are specified in the modification plist, the existing values are retained. If any subject's distinguished name fields are specified, the name is completely replaced by the values specified.
24. For REQDETAILS the subject name fields are not returned in the CertPlist if the subject distinguished name does not conform to RACF name standards regarding RDN qualifiers and order. See usage note 15 for more information.
25. SAF return code 8, RACF return code 8, RACF reason code 4 indicates a problem with the value specified for either the Number_parameters or Attributes parameter.
26. The R_PKIServ callable service creates SMF type 80 records, with event codes of 69, 70, 72, 73, and 74. RACF audits the invocations of this callable service under the following circumstances:
 - a. UAUDIT is in effect for the user
 - b. The user has SPECIAL authority and SETROPTS SAUDIT is in effect
 - c. The request is successful and SETROPTS AUDIT(USER) is in effect
 - d. The request fails due to insufficient authorization

For details on the SMF records produced see *z/OS Security Server RACF Macros and Interfaces*.

27. For GENCERT and REQCERT, if the CertPlist fields Email and NotifyEmail are specified together, they must have the same value. Otherwise, the service will fail with reason code 76. For GENRENEW and REQRENEW, if the CertPlist field NotifyEmail is specified and the subject's distinguished name being renewed contains the MAIL attribute, the two values must be the same. Otherwise, the service will fail with reason code 76. In either case "NotifyEmail" is returned in the DiagInfo area as the field name in error.
28. For MODIFYREQS, if a modification parameter list (CertPlist) is specified and it contains the field Email and NotifyEmail was specified on the original request, the new Email value will replace the old NotifyEmail value.

R_proxyserv (IRRSPY00): LDAP interface

Function

The **R_proxyserv** SAF callable service invokes the LDAP component of the Security Server for z/OS to obtain data which resides in an LDAP directory. Invokers are not required to be LE-enabled.

Requirements

Authorization:	Any PSW key in supervisor or problemstate
Dispatchable unit mode:	Task of user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have a FRR active
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. The words containing the ALETs must be in the primary address space.

Linkage Conventions

The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, the last word in the parameter list must have a 1 in the high-order (sign) bit.

RACF Authorization

For callers not running in system key or supervisor state, the use of **R_proxyserv** is authorized by the resource **IRR.RPROXYSERV** in the **FACILITY** class. The application server must be running with a RACF user or group ID that has at least **READ** authority to this resource. If the class is inactive, or the resource is not defined, only servers running with a system key or in supervisor state may use the **R_proxyserv** service.

Format

```
CALL IRRSPY00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ParmALET,
               Function_code,
               LDAP_host,
               Bind_DN,
               Bind_PW,
               Host_userID,
               Base_DN,
               Result_entries
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF. The work area must be in the primary address space.

ALET

The name of a fullword which must be in the primary address space and contains the ALET for the remaining parameters.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

ParmALET

The name of a fullword containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

Function_code

The name of a half word (2 byte) area containing the Function code. The function code has one of the following values:

X'0001'	Return the distinguished name (DN) of the user identified by the input host user ID.
X'0002'	Return the Policy Director Authorization Services attributes for the specified base DN.

LADP_host

This is optional, see Usage Notes. The name of an area that consists of a 4 byte length field followed by the string of EBCDIC characters which identify the URL of the LDAP server which the z/OS LDAP Server is to contact when acting as a proxy for this request. The maximum length of this string is 1023 bytes. Uppercase and lowercase characters are allowed, but no significance is attached to the case. See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP URLs and the *z/OS Security Server RACF Command Language Reference* for more information about how to define this information in the RACF database.

Bind_DN

This is optional, see Usage Notes. The name of an area that consists of a 4 byte length field followed by the string of EBCDIC characters which represent the distinguished name (DN) that the z/OS LDAP Server is to use when acting as a proxy for this request. The maximum length of this string is 1023 bytes. Both uppercase and lowercase characters are allowed. See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP distinguished names and the *z/OS Security Server RACF Command Language Reference*, for more information about how to define this information in the RACF database.

Bind_PW

This is optional, see Usage Notes. The name of an area that consists of a 4 byte length field followed by the string of EBCDIC characters which represent the password that the z/OS LDAP Server is to use when acting as a proxy for this request. The maximum length of this string is 128 bytes. Both uppercase and lowercase characters are allowed. See *z/OS Security Server LDAP Server*

Administration and Use for more information about LDAP passwords and the ,
z/OS Security Server RACF Command Language Reference for more information
 about how to define this information in the RACF database.

Host_userID

The name of a 9 byte area that consists of a 1 byte length field followed by up to 8 EBCDIC characters. Uppercase and lowercase characters are allowed, but no significance is attached to the case. If not specified, the length must equal 0.

Base_DN

The name of an area that consists of a 4 byte length field followed by the string of EBCDIC characters which represent the base DN of an LDAP subtree. The maximum length of this string is 1023 bytes.

Result_entries

The name of a 4 byte area which contains the address of the output area from this service, if any. All character fields in the output area are represented in EBCDIC. The caller is responsible for releasing output area storage, via FREEMAIN or STORAGE RELEASE macro invocation. Offset values in the output area are all relative to the beginning of the output area.

Table 99. Result_entries output area

OFFSET (DECIMAL)	OFFSET (HEX)	TYPE	LENGTH	NAME	DESCRIPTION
0	0	STRUCTURE	28	ResultArea	ResultArea
0	0	UNSIGNED	4	ResultAreaLen	Length of results area
4	4	UNSIGNED	1	ResultAreaVersion	Format Version
5	5	UNSIGNED	1	ResultAreaSubpool	Storage subpool
6	6	CHARACTER	22	*	Reserved
28	1C	CHARACTER		ResultAreaData	Function code specific result data

When the function code is X'0001', Return Distinguished Name (DN), the function code specific result data (ResultAreaData) will be mapped as follows:

OFFSET (DECIMAL)	OFFSET (HEX)	TYPE	LENGTH	NAME	DESCRIPTION
0	0	STRUCTURE	8	DNData	DN result data
0	0	UNSIGNED	4	DNNumber	Number of DNs returned
4	4	UNSIGNED	4	DNList@	Offset to start of DN array

OFFSET (DECIMAL)	OFFSET (HEX)	TYPE	LENGTH	NAME	DESCRIPTION
0	0	STRUCTURE	8	DNList(*)	DN array
0	0	UNSIGNED	4	DNLen	Length of DN
4	4	UNSIGNED	4	DN@	Offset to DN

R_proxyserv

When the function code is X'0002', Return Policy Director Authorization Services Attributes, the function code specific data (ResultAreaData) will be mapped as follows:

OFFSET (DECIMAL)	OFFSET (HEX)	TYPE	LENGTH	NAME	DESCRIPTION
0	0	STRUCTURE	52	PrivilegeData	Privilege result data
0	0	UNSIGNED	1	PrivPasswordValid	Boolean
1	1	UNSIGNED	1	PrivAccountValid	Boolean
2	2	UNSIGNED	2	*	Reserved
4	4	UNSIGNED	4	PrivDomainNameLen	Length of domain name
8	8	UNSIGNED	4	PrivDomainName@	Offset to domain name
12	C	UNSIGNED	4	PrivLoginNameLen	Length of login name
16	10	UNSIGNED	4	PrivLoginName@	Offset to login name
20	14	UNSIGNED	4	PrivPrincipalNameLen	Length of principal
24	18	UNSIGNED	4	PrivPrincipalName@	Offset to principal name
28	1C	UNSIGNED	4	PrivUserNameLen	Length of user name
32	20	UNSIGNED	4	PrivUserName@	Offset to user name
36	24	UNSIGNED	4	PrivUserUUIDLen	Length of UUID
40	28	UNSIGNED	4	PrivUserUUID@	Offset to UUID
44	2C	UNSIGNED	4	PrivNumberGroups	Number of groups
48	30	UNSIGNED	4	PrivGroups@	Offset to start of group array

OFFSET (DECIMAL)	OFFSET (HEX)	TYPE	LENGTH	NAME	DESCRIPTION
0	0	STRUCTURE	16	PrivGroups(*)	Group array
0	0	UNSIGNED	4	PrivGroupNameLen	Length of group name
4	4	UNSIGNED	4	PrivGroupName@	Offset to group name
8	8	UNSIGNED	4	PrivGroupUUIDLen	Length of group UUID
12	C	UNSIGNED	4	PrivGroupUUID@	Offset to group UUID

Return and Reason Codes

IRRSPY00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
4	0	0	RACF is not installed.
8	8	0	Invalid function code.
8	8	4	Parameter list error.
8	8	8	An internal error was encountered. A record may be written to LOGREC with more diagnostic information.
8	8	12	A recovery environment could not be established.
8	8	16	Not authorized to use this service.
8	8	20	The z/OS LDAP Server address space has not been started or is terminating.
8	8	24	Unable to decode the data retrieved from LDAP. A record is written to LOGREC with more diagnostic information.
8	8	28	Unable to determine BIND information for LDAP.
8	12	8	z/OS LDAP Server invocation failed-parameter buffer overflow. This indicates an internal error in IRRRPY00.
8	12	12	z/OS LDAP Server invocation failed-unable to allocate storage. The region size for the z/OS LDAP Server program should be increased.
8	12	16	z/OS LDAP Server invocation failed-LDAP PC handling is not enabled.
8	12	20	z/OS LDAP Server invocation failed-abend in the PC service routine. The symptom record associated with this abend can be found in LOGREC.
8	12	24	z/OS LDAP Server invocation failed-no control area (internal error). Report the problem to the IBM support center and provide an SVC dump of the z/OS LDAP Server address space.
8	12	36	z/OS LDAP Server invocation failed-the z/OS LDAP Server is busy. Retry the operation.
8	12	40	z/OS LDAP Server invocation failed-PC request processing was terminated before completion (the PC catcher initiated the termination). This indicates either an internal error in IRRRPY00 or that the z/OS LDAP Server is terminating. If the z/OS LDAP Server is terminating, restart it and retry the operation.

R_proxyserv

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
8	23	44	z/OS LDAP Server invocation failed-PC request processing was terminated before completion (the server agent initiated the termination). This indicates either an internal error in IRRRPY00 or that the z/OS LDAP Server is terminating. If the z/OS LDAP Server is terminating, restart it and retry the operation.
8	12	52	z/OS LDAP Server invocation failed-a lock could not be obtained. This indicates either that the z/OS LDAP Server is busy or that an internal error is preventing it from processing requests. If the z/OS LDAP Server is not busy, report the problem to the IBM support center and provide an SVC dump of the z/OS LDAP Server address space.
8	16	nn	The requested function was not successful. The RACF reason code code is set to the return code from LDAP. Please refer to <i>z/OS Security Server LDAP Server Administration and Use</i> for more information about LDAP return codes. For reason codes other than 32, a record is written to LOGREC containing the LDAP reason string. No record is written for reason code 32, which indicates that the requested information does not exist in the LDAP directory.
8	20	40	Invalid BIND password information was found in either the PROXY segment of the invoker's USER class profile, or in the PROXY segment of the IRR.PROXY.DEFAULTS profile in the FACILITY class.
8	20	44	RACF was unable to retrieve or update the master key/master key token in the SSIGNON segment of the LDAP.BINDPW.KEY profile in the KEYSMSTR class.
8	20	52	RACF cannot locate the CCA support routine.
8	20	56	The invocation of the CCA support routine has failed.

Parameter Usage

Function	Function Code	LDAP_host	Bind_DN	Bind_PW	Host_userID	Base_DN	Result_entries
Return the base DN for the specified host UID	X'0001'	In ¹	In ¹	In ¹	In	N/A	Out

Function	Function Code	LDAP_host	Bind_DN	Bind_PW	Host_userID	Base_DN	Result_entries
Return the Policy Director attributes for the specified base DN	X'0002'	In ¹	In ¹	In ¹	N/A	In	Out

¹Parameter is optional, see Usage Notes.

Usage Notes

1. This service is intended for use by z/OS application servers, which are not executing in a Language Environment (LE). It allows z/OS application servers to perform limited LDAP queries which retrieve information from a directory information tree (DIT). Note, however, that LE-enabled applications may also exploit this service, should they choose to do so.
2. The R_proxyserv service requires an instance of the z/OS LDAP Server or OS/390 LDAP Server on each physical z/OS or OS/390 instance (whether in a sysplex datasharing configuration or not) and each of these LDAP Server instances must be configured to support PC call and the extended operations backend. See the *z/OS Security Server LDAP Server Administration and Use* for information about configuring this support.
3. The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, the last word in the parameter list must have a 1 in the high-order (sign) bit of the last word in the parameter list. If the last word in the parameter list does not have a 1 in the high-order (sign) bit, the caller receives a parameter list error. The first parameter that can have the high-order bit on, ending the parameter list, is the *Result_entries* parameter.
4. The LDAP_host, Bind_DN, and Bind_PW parameters are all optional. If any of the three parameters are specified, all must be specified, or R_proxyserv will return an error. If all three parameters are omitted, RACF attempts to determine this information from the PROXY segment associated with the RACF user identity of the invoker (i.e. the server's address space level ACEE). If the user profile PROXY segment is found, but any of the corresponding segment values (LDAPHOST, BINDDN, or BINDPW) are not defined, R_proxyserv will return an error. If the LDAP_host, Bind_DN, and Bind_PW parameters are omitted and the PROXY segment is not defined for the invoker's user identity, RACF will then look for the IRR.PROXY.DEFAULTS profile in the FACILITY class. If this profile is not found or does not have a PROXY segment or does not have values defined for LDAPHOST, BINDDN, and BINDPW, R_proxyserv will return an error.
5. The format of the Result_entries output area will differ, based upon the function code specified. Mappings are provided for each format (see Mappings for Result_entries output area). Storage will be obtained in primary in the subpool indicated in the Result_entries output area and it is the responsibility of the invoker to release this storage.

Related Services

None

R_ptrace (IRRSPT00): Ptrace Authority Check

Function

The **R_ptrace** service checks whether the calling process can ptrace the target process.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. **R_ptrace** checks whether the caller is a superuser or whether the caller is the owner of the target process, and verifies that the target process is not running a SETUID or SETGID program.
2. If the caller is not superuser nor the process owner, an authorization check is performed on the resource name in the UNIXPRIV class shown in Table 100. If the authorization check is successful, the caller is treated as a superuser.

Table 100. UNIXPRIV class resource names used in R_ptrace

Audit function code	Resource name	Access required
N/A	SUPERUSER.PROCESS.PTRACE	READ

Format

```
CALL IRRSPT00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Target_process_UIDs,
               ALET, Target_process_GIDs,
               ALET, Target_PID
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each

parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Target_process_UIDs

The address of a 3-word area containing the real, effective, and saved z/OS UNIX user identifiers (UIDs) (in that order) for the target process.

Target_process_GIDs

The address of a 3-word area containing the real, effective, and saved z/OS UNIX group identifiers (GIDs) (in that order) for the target process.

Target_PID

The name of a fullword containing the PID of the target process.

Return and Reason Codes

IRRSP00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The caller is not authorized to ptrace the target process.
8	8	12	An internal error occurred during RACF processing.

Usage Notes

1. This service is intended only for use by the MVS BCP.
2. An audit record is optionally written, depending on the audit options in effect for the system.
3. This service uses task level support when z/OS UNIX has indicated in the task's ACEE that this is a task level process.

Related Services

None

R_setegid (IRRSEG00): Set Effective GID, Set All GIDs

Function

The **R_setegid** service checks whether the user is authorized to change the GID and, if so, changes the effective GID for the current process.

If the high-order bit of the input GID is on, the real, effective, and saved GIDs are changed for the current process.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN= HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. If the high-order bit of the input GID is off and if the user is the superuser or if the input GID is equal to the real or saved GID of the calling process, the effective GID of the process is changed to the input GID. The real and saved GIDs are not changed. The new values of the GIDs are returned to the calling process.

Format

```
CALL IRRSEG00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, GID,
               ALET, Output_area
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

GID

The name of a fullword containing the GID to be set. The GID must be defined

to RACF. If the high-order bit is on, the GIDs stored in the output area are stored as the real, effective, and saved GIDs (in that order) for the current process.

Output_area

The name of a 3-word area in which the new real, effective, and saved GIDs (in that order) are returned. If the high-order bit of the GID is on, the real, effective, and saved GIDs in this area are stored as the real, effective, and saved GIDs for the current process.

Return and Reason Codes

IRRSEG00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The GID is not defined to RACF.
8	8	8	The user is not authorized to change the GID.
8	8	12	An internal error occurred during RACF processing.
8	8	16	Recovery could not be established.

Usage Notes

1. This service is intended only for use by the MVS BCP.
2. IRRSEG00 only changes the GIDs. The user's current group in the ACEE is not changed. Therefore, IRRSEG00 only affects access to z/OS UNIX files. Access to other MVS files is not changed.
3. An audit record is written.

Related Services

None

R_seteuid (IRRSEU00): Set Effective UID, Set All UIDs

Function

The **R_seteuid** service checks whether the user is authorized to change the z/OS UNIX user identifiers (UIDs) and, if so, changes the effective UID for the current process.

If the high-order bit of the input UID is on, the real, effective, and saved UIDs are changed for the current process.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN= HASN
AMODE:	31

R_seteuid

RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. If the high-order bit of the input z/OS UNIX user identifier (UID) is off and if the user is the superuser or if the input UID is equal to the real or saved UID of the calling process, the effective UID of the process is changed to the input UID. The real and saved UIDs are not changed. The new values of the UIDs are returned to the calling process.

Format

```
CALL IRRSEU00 (Work_area,  
              ALET, SAF_return_code,  
              ALET, RACF_return_code,  
              ALET, RACF_reason_code,  
              ALET, UID,  
              ALET, Output_area  
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

UID

The name of a fullword containing the z/OS UNIX user identifier (UID) to be set. The UID must be defined to RACF. If the high-order bit is on, the UIDs stored in the output area are stored as the real, effective, and saved UIDs (in that order) for the current process.

Output_area

The name of a 3-word area in which the new real, effective, and saved z/OS UNIX user identifiers (UIDs) (in that order) are returned. If the high-order bit

of the UID is on, the real, effective, and saved UIDs in this area are stored as the real, effective, and saved UIDs for the current process.

Return and Reason Codes

IRRSEU00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The z/OS UNIX user identifier (UID) is not defined to RACF.
8	8	8	The user is not authorized to change the z/OS UNIX user identifier (UID).
8	8	12	An internal error occurred during RACF processing
8	8	16	Recovery could not be established.

Usage Notes

1. This service is intended only for use by the MVS BCP.
2. For additional security-related information, see the description of the **seteuid** callable service in *z/OS UNIX System Services Programming: Assembler Callable Services Reference*
3. An audit record is written.

Related Services

None

R_setfac1 (IRRSCLO0):Unix Access Control Lists

Function

The **R_setfac1** callable service is used to maintain the access lists for a UNIX file or directory

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	SETFRR
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF authorization

To change the ACL, the user must be a superuser or must be the owner of the file. To be considered a superuser, the user must either have a UID value of 0, or must be permitted with at least READ access to the resource named SUPERUSER.FILESYS.CHANGEPERMS in the UNIXPRIV class.

Format

```
CALL IRRSCL00 (Work_area,
              ALET, SAF_return_code,
              ALET, RACF_return_code,
              ALET, RACF_reason_code,
              ALET, ACL_Update,
              ALET, ACL_Update_length,
              ALET, FSP,
              ALET, File_identifier,
              ALET, CRED,
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

ACL_Update

The name of an area containing the type of ACL being updated, the operation being requested, and an ACL structure which contains entries to be added, updated, or removed. See the RACL_Edit structure in the IRRPCOMP macro for the mapping of this area.

The ACL structure is mapped by IRRPFACL. If the operation is add and entries are specified in this ACL mapping, then the current ACL will be replaced with the entries specified in this structure.

ACL_Update_Length

The name of a fullword containing the length of the ACL_Update buffer.

FSP

The name of the IFSP for the file whose mode bits are to be changed.

File_Identifier

The name of a 16-byte area containing a unique identifier of the file.

CRED

The name of the CRED structure for the current file system syscall. See z/OS

Security Server RACF Data Areas for more information. The CRED contains a pointer to the ACL being modified/deleted, or contains the address of a buffer in which to create a new ACL.

Return and reason codes

IRRSC100 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The user is not authorized to change the ACL.
8	8	8	The maximum of 1024 entries would be exceeded by this request.
8	8	12	An internal error occurred during RACF processing.
8	8	16	An error was encountered in the ACL passed in the ACL_Update parameter. FACL_ErrOff contains the offset to the header field or the ACL entry in error. See usage notes for possible error conditions.
8	8	20	ACL buffer provided by caller is not large enough to contain a valid ACL
8	8	24	Input parameter list error.
8	0	32	The CRED user type is not supported.

Usage notes

1. This service is intended only for use by an z/OS UNIX System Services file system and by z/OS UNIX System Services servers. The service contains support for z/OS UNIX System Services servers, but cannot be directly invoked by an z/OS UNIX System Services server.
2. An access list may contain a maximum of 1024 entries.
3. R_setfac1 will manage the bit in the File Security Packet (FSP) which indicates the presence of an ACL of a given type. That is, when an ACL is successfully added (via either the add or modify operation), R_setfac1 will turn on the appropriate bit in IFSP_FLAG2 (either IFSP_Access_Acl, IFSP_File_Model_Acl, or IFSP_Dir_Model_Acl). For a delete operation, or an add or modify operation which results in an empty ACL, RACF will turn off the appropriate bit in IFSP_FLAG2 .
4. When a modify operation is specified, requests to delete ACL entries are processed before requests to add or modify entries.
5. If a modify operation is specified and an ACL does not already exist, it will be created. Likewise, if a modify for a specific ACL entry is specified, and that entry does not already exist, it will be created.
6. If a delete request is specified, but an ACL does not already exist, the request will be ignored. Likewise, if a delete of a specific ACL entry is specified, and that entry does not already exist, it will be ignored.

R_setfac1

7. If an add request is specified, and an ACL already exists, it will be replaced in accordance with the contents of the RACL_Edit structure pointed to by the ACL_Update parameter. If there is no RACL_Edit in this context, the existing ACL will be deleted.
8. If a delete request is specified, and a RACL_Edit structure is also contained within the structure pointed to by the ACL_Update parameter, then the RACL_Edit is ignored and the ACL is deleted.
9. An audit record (or records) is optionally written, depending on the audit options in effect for the system.
10. The parameter list passed to this service is a variable-length (VL) parameter list. The high-order bit of the last field must be set to mark the end of the parameter list.
11. The caller must pass in the length and address of a buffer which contains the ACL being modified, or in which a new ACL is to be created. The buffer must be large enough to contain the maximum size ACL. The length and address fields are contained within the CRED, and different field names are used depending on which ACL is being created, modified, or deleted. For an access ACL, use CredAccAcl and CredAccAclLen. For a directory model ACL, use CredDirModelAcl and CredDirModelAclLen. For a file model ACL, use CredFileModelAcl and CredFileModelAclLen.
12. R_setfac1 will perform validation on the ACL passed into the service as part of the RACL_Edit parameter of IRRPCOMP. An error in this ACL will result in a SAF return code 8, RACF return code 8, and RACF reason code 16 (decimal). If an error is detected, the FACL_ErrOff field within this ACL mapping will be updated with the offset (from the start of the header) to the header field or ACL entry in error. Some of the items validated are: eye catcher = "FACL", version = 1, length is large enough to contain the number of entries specified in FACL_Num_Entry, the ACL contains at least one entry, ACL entry type is 1 or 2, and UID/GID value is greater than or equal to 0.
13. An error with the input parameter list will result in a SAF return code 8, RACF return code 8, and RACF reason code 24 (decimal). Some of the items validated are: all addresses in the parameter list are non-zero, the variable-length parameter list bit is set, the ACL_Update_Length parameter specifies a length which is large enough to contain the ACL_Update area, the operation type and ACL type specified in the ACL_Update area are valid, and the pointers in the CRED which point to ACL buffers are non-zero and point to an area which is large enough to contain the ACL.

Related services

R_chmod, R_chown, ck_access

R_setgid (IRRSSG00): Set Group Name

Function

The **R_setgid** service checks whether the user is authorized to change the GIDs and, if so, changes the real, saved, or effective GID (or some combination of these) for the current process.

Requirements

Authorization:

Any PSW key in supervisor state

Dispatchable unit mode:

Task of z/OS UNIX user

Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. If the calling process is a superuser, the real, saved, and effective GIDs are changed. If the calling process is not a superuser but the input GID is equal to the real or saved GID, the process's effective GID is changed. If neither condition is met, the process's GIDs are not changed, and an error return code and an error reason code are returned.

Format

```
CALL IRRSSG00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, GID,
               ALET, Output_area
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

GID

The name of a fullword containing the GID to be set. The GID must be defined to RACF.

Output_area

The name of a 3-word area in which the new real, effective, and saved GIDs (in that order) are returned.

R_setgid

Return and Reason Codes

IRRSSG00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The GID is not defined to RACF.
8	8	8	The user is not authorized to change the GID.
8	8	12	An internal error occurred during RACF processing.
8	8	16	Recovery could not be established.

Usage Notes

1. This service is intended only for use by the MVS BCP.
2. IRRSSG00 changes only the GIDs. No change is made to the user's current group in the ACEE. Therefore, IRRSSG00 only affects access to z/OS UNIX files. Access to other MVS files is unchanged.
3. An audit record is written.

Related Services

None

R_setuid (IRRSSU00): Set z/OS UNIX user identifier (UID)

Function

The **R_setuid** service checks whether the user is authorized to change the z/OS UNIX user identifiers (UIDs) and, if so, changes the real, saved, or effective UID (or some combination of these) for the current process.

Requirements

Authorization:	Any PSW key in supervisor state
Dispatchable unit mode:	Task of z/OS UNIX user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

1. If the calling process is a superuser, the real, saved, and effective z/OS UNIX user identifiers (UIDs) are changed. If the calling process is not a superuser, but the input UID is equal to the real or saved UID, the process's effective UID is changed. If neither condition is met, the process's UIDs are not changed, and an error return code and an error reason code are returned.

Format

```
CALL IRRSSU00 (Work_area,
               ALET,SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, UID,
               ALET, Output_area
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

UID

The name of a fullword containing the z/OS UNIX user identifier (UID) to be set. The UID must be defined to RACF.

Output_area

The name of a 3-word area in which the new real, effective, and saved z/OS UNIX user identifiers (UIDs) (in that order) are returned.

Return and Reason Codes

IRRSSU00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	The z/OS UNIX user identifier (UID) is not defined to RACF.
8	8	8	The user is not authorized to change the z/OS UNIX user identifier (UID).

R_setuid

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
8	8	12	An internal error occurred during RACF processing.
8	8	16	Recovery could not be established.

Usage Notes

1. This service is intended only for use by the MVS BCP.
2. For additional security-related information, see the description of the **setuid** callable service in *z/OS UNIX System Services Programming: Assembler Callable Services Reference*.
3. An audit record is optionally written, depending on the audit options in effect for the system.

Related Services

None

R_ticketserv (IRRSPK00): Parse or Extract

Function

The **R_ticketserv** service enables z/OS application servers to parse or extract principal names from a GSS-API context token. This enables an z/OS application server to determine the client principal who originated an application-specific request, when the request includes a GSS-API context token and the intended recipient is the z/OS application server. For more information on the GSS-API services supported on z/OS, see the Security Server Network Authentication Service product documentation.

Requirements

Authorization:	Any PSW key in supervisor state or problem state
Dispatchable unit mode:	Task of user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. The words containing the ALETs must be in the primary address space.

Linkage Conventions

The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, the last word in the parameter list must have a one in the high-order (sign) bit.

RACF Authorization

For servers not running in system key or supervisor state, the use of R_ticketerv service is authorized by the resource IRR.RTICKETSERV in the FACILITY class. The application server must be running with a RACF user or group ID that has at least READ authority to this resource. If the class is inactive, or the resource is not defined, only servers running system key or supervisor state may use the R_ticketerv service.

Format

```
CALL IRRSPK00 (Work_area,
              ALET, SAF_return_code,
              ALET, RACF_return_code,
              ALET, RACF_reason_code,
              ALET, Function_code,
                  Option_word,
                  Ticket_area,
                  Ticket_options,
                  Ticket_principal_userid
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a full word in which the SAF router returns the SAF return code.

RACF_return_code

The name of a full word in which the service routine stores the return code.

RACF_reason_code

The name of a full word in which the service routine stores the reason code.

ALET

The name of a word which must be in the primary address space and which contains the ALET for the following fields:

- Function_code
- Option_word
- Ticket_area
- Ticket_options
- Ticket_principal_userid

Function_code

The name of a half word (2 byte) area containing the Function code. The function code has one of the following values:

X'0001'

Parse specified ticket and return Network Authentication Service principal name.

Option_word

The name of a fullword containing binary zeros. The area pointed to by this parameter is reserved for future use.

R_ticketserv

Ticket_area

The name of an area that consists of a 2 byte field, followed by the GSS-API context token. For function code 1, extract Network Authentication Service V5 principal, the GSS-API context token is assumed by SAF to have been created exclusively by the Network Authentication Service mechanism.

Ticket_options

The address of a binary bit string which identifies the ticket specific processing to be performed. This parameter is reserved for future use.

Ticket_principal_userid

The name of a 242 byte area that consists of a 2 byte length field followed by the name of the Ticket principal user ID.

Note: Fully qualified Network Authentication Service names will be returned, using a case sensitive, DCE-like naming convention:
/.../realm_name/principal_name

Return and Reason Codes

IRRSPK00 may return the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	0	Invalid function code.
8	8	4	Parameter list error.
8	8	8	An internal error was encountered.
8	8	12	A recovery environment could not be established.
8	8	16	Not authorized to use this service.
8	12	8	Invocation of the Security Server Network Authentication Service Program Call (PC) interface failed with a 'parameter buffer overflow' return code. This indicates an internal error in IRRSPK00.
8	12	12	Invocation of the Security Server Network Authentication Service Program Call (PC) interface failed with an 'unable to allocate storage' return code. The region size for the Security Server Network Authentication Service started task (SKRBKDC) should be increased.
8	12	16	Invocation of the Security Server Network Authentication Service Program Call (PC) interface failed with a 'local services are not available' return code. This indicates that the Security Server Network Authentication Service started task (SKRBKDC) address space has not been started or is terminating.

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
8	12	20	Invocation of the Security Server Network Authentication Service Program Call (PC) interface failed with a 'abend in the PC service routine' return code. The symptom record associated with this abend can be found in the logrec dataset.
8	12	24	Invocation of the Security Server Network Authentication Service Program Call (PC) interface failed with an 'unable to obtain control lock' return code. This can occur if the task holding the lock is not being dispatched (for example, a dump is in progress).
8	16	X'nnnnnnnn'	The Security Server Network Authentication Service was not able to successfully extract the client principal name from the supplied Kerberos V5 ticket. X'nnnnnnnn' is the Kerberos return code. Refer to the Security Server Network Authentication Service documentation for more information.

Usage Notes

1. This service is intended for use by z/OS application servers. It allows application servers with a Kerberos V5 ticket to determine the Kerberos principal associated with the ticket.
2. This service requires that the Security Server Network Authentication Service be installed and executing. Otherwise, SAF return code 8, RACF return code 12, and RACF reason code 16 will be returned to the invoker.
3. In a datasharing sysplex, there must be an Security Server Network Authentication Service instance running on each system in the sysplex. The Security Server Network Authentication Service instances must all be in the same realm and share the same RACF database (if they do not share the same database, then they cannot be in the same realm).
4. An ALET must be specified for the SAF_return_code, RACF_return_code, and RACF_reason_code parameters, and a single ALET specified for all of the remaining parameters.
5. The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, the last word in the parameter list must have a one in the high-order (sign) bit. If the last word in the parameter list does not have a one in the high-order (sign) bit, the caller receives a parameter list error. The first parameter that can have the high-order bit on, ending the parameter list, is the Ticket_principal_userid parameter.
6. The calling z/OS application server must have a local Kerberos identity defined as a populated KERB segment of the server's RACF User profile.
7. For function code X'0001', a SAF return code 8 and a RACF return code 16 indicates that the Security Server Network Authentication Service was unable to successfully process the input Kerberos V5 ticket. The return code is passed back to the invoker as the RACF reason code. Some of the more common return codes are the following:

R_ticketserv

- X'861B6D04' (G_BUFFER_ALLOC)=storage not available for GSS-API control block.
- X'861B6D06' (G_WRONG_SIZE)=client principal name is too long for result buffer.
- X'861B6D0B' (G_BAD_TOK_HEADER)=the GSS-API token header is incorrect.
- X'861B6D58' (G_UNEXPECTED_TOKEN)=the GSS-API token was not created by the gss_init_sec_context() function.
- X'861B6D60' (G_UNSUPPORTED_MECHANISM)=unsupported GSS-API security mechanism.
- X'96C73A07'(KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN)=the current RACF userid is not associated with a Kerberos principal.
- X'96C73A20'(KRB5KDC_AP_ERR_TKT_EXPIRED)=Kerberos ticket is expired.
- X'96C73A25'(KRB5KDC_AP_ERR_SKEW)=Client and server clocks are not synchronized or authenticator is expired.
- X'96C73A90'(KRB5KDC_AP_WRONG_PRINC)=the server principal in the GSS-API security token does not match the principal associated with the current RACF userid.
- X'96C73C02'(KRB5_NOMEM)=storage not available for Kerberos control block.

Parameter Usage

Parameter	Function Code X'0001'
SAF_return_code	Output
RACF_return_code	Output
RACF_reason_code	Output
Function_code	Input
Option_word	Reserved
Ticket_area	Input
Ticket_options	Reserved
Ticket_principal_userid	Output

Related Services

R_kerbinfo, R_usermap

R_umask (IRRSMM00): Set File Mode Creation Mask

Function

The **R_umask** service sets the file mode creation mask for the current process to the permission bits specified in the input mode parameter. It returns the permission bits that were in the file mode creation map in the mode parameter.

Requirements

Authorization: Any PSW key in supervisor state
Dispatchable unit mode: Task of z/OS UNIX user
Cross memory mode: PASN = HASN or PASN not = HASN

AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	None
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

RACF Authorization

None

Format

```
CALL IRRSMM00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Mode
               )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Mode

The name of a word containing the mode bits to be set in file mode creation mask of the current process. Only the file permission bits in the mode parameter are used. Other defined bits are ignored.

On output, the mode word is zeroed and the permission bits that were in the file mode creation mask are set in the mode word.

See "File Type and File Mode Values" on page 4 for a definition of the security bits in the mode parameter.

Return and Reason Codes

IRRSMM00 returns the following values in the reason and return code parameters:

SAF Return Code	RACF Return Code	RACF Reason Code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.

Usage Note

- This service is intended only for use by the MVS BCP and by z/OS UNIX servers. The service contains support for z/OS UNIX servers, but can not be directly invoked by a z/OS UNIX server.

Related services

chmod, makeFSP

R_usermap (IRRSIM00): Map application user

Function

The **R_usermap** service enables z/OS application servers to determine the application user identity associated with a RACF user ID, or to determine the RACF user ID associated with an application user identity or digital certificate. Examples of applications supported are RACF user ID, application user identity, application, Lotus Notes for z/OS and Novell Directory Services (NDS).

This service can only map application user identities which have already been defined to RACF:

- For Lotus Notes for z/OS, the RACF USER profile must have an LNOTES segment containing a short name. This can be added with the ADDUSER or ALTUSER command, or the R_admin callable service.
- For NDS for z/OS, the RACF USER profile must have an NDS segment containing a user name. This can be added with the ADDUSER or ALTUSER command, or the R_admin callable service.
- For digital certificates, the certificate must be associated with a RACF user ID through automatic registration or with the RACDCERT command.
- For Security Server Network Authentication Service, local Kerberos principals require a RACF USER profile with a KERB segment containing a principal name. Foreign Kerberos principals must be defined to RACF using KERBLINK profiles.

Requirements

Authorization:	Any PSW key in supervisor or problem state
Dispatchable unit mode:	Task of user
Cross memory mode:	PASN = HASN
AMODE:	31
RMODE:	Any
ASC mode:	Primary or AR mode
Recovery mode:	ESTAE. Caller cannot have an FRR active.
Serialization:	Enabled for interrupts
Locks:	No locks held

Control parameters:

The parameter list and the work area must be in the primary address space. The words containing the ALETs must be in the primary address space.

Linkage Conventions

The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, the last word in the parameter list must have a 1 in the high-order (sign) bit.

RACF Authorization

For servers not running in system key or supervisor state, the use of the R_usermap service is authorized by the resource **IRR.RUSERMAP** in the **FACILITY** class. The application server must be running with a RACF user ID or group ID that has at least **READ** authority to this resource. If the class is inactive, or the resource is not defined, only servers running in system key or supervisor state may use the R_usermap service.

Format

```
CALL IRRSIM00 (Work_area,
              ALET, SAF_return_code,
              ALET, RACF_return_code,
              ALET, RACF_reason_code,
              ALET, Function_code,
              Option_word,
              RACF_userid,
              Certificate,
              Application_userid
              )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space and must be on a double word boundary.

ALET

The name of a word containing the ALET for the following parameter. Each ALET can be different. The last ALET in the parameter list will be used for the remainder of the parameters. The words containing the ALETs must be in the primary address space.

SAF_return_code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_return_code

The name of a fullword in which the service routine stores the return code.

RACF_reason_code

The name of a fullword in which the service routine stores the reason code.

Function_code

The name of a halfword containing the function code. The function code has one of the following values:

R_usermap

X'0001'

Return the Lotus Notes for z/OS application user identity associated with the supplied RACF user ID.

X'0002'

Return the RACF user ID associated with the supplied Lotus Notes for z/OS application user identity or digital certificate.

X'0003'

Return the NDS for z/OS application user identity associated with the supplied RACF user ID.

X'0004'

Return the RACF user ID associated with the supplied NDS for z/OS application user identity or digital certificate.

X'0005'

Return the Network Authentication Service application user identity associated with the supplied RACF user ID.

Note: This functions only with local Network Authentication Service principals.

X'0006'

Return the RACF user ID associated with the supplied Network Authentication Service application user identity or digital certificate.

Option_word

The name of a fullword containing binary zeros. The area pointed to by this parameter is reserved for future use.

RACF_userid

The name of a 9 byte area that consists of a 1 byte length field followed by up to 8 characters. It must be specified in upper case. If not specified, the length must equal 0.

Certificate

The name of an area that consists of a 4 byte length field followed by a digital certificate. The certificate must be a single BER encoded X.509 certificate. If not specified, the length must equal 0.

Application_userid

The name of a 248 byte area that consists of a 2 byte length field followed by the name of the application user identity. If not specified, the length must equal 0.

Return and Reason Codes

R_usermap may return the following values in the reason and return code parameters:

SAF return code	RACF return code	RACF reason code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	Parameter list error occurred.
8	8	8	An internal error occurred during RACF processing.

SAF return code	RACF return code	RACF reason code	Explanation
8	8	12	Recovery environment could not be established.
8	8	16	There is no mapping between RACF and an application. For function codes 1 and 3, and 5, the RACF user ID exists but there is either no SNAME in the LNOTES segment or no LNOTES segment, or there is no UNAME in the NDS segment or no NDS segment, or no KERBNAME in the KERB segment, or no KERB segment. For function codes 2, 4, and 6, there is no mapping to a RACF user ID for the application identity provided.
8	8	20	Not authorized to use this service.
8	8	24	The specified RACF user ID does not exist.
8	8	28	Certificate is not valid.
8	8	32	Either no RACF user ID is defined for this certificate, or the certificate status is NOTRUST.

Parameter Usage

Table 101. Parameter Usage

Parameter	Function Code 1 (RACF to Notes)	Function Code 2 (Notes to RACF)	Function Code 3 (RACF to NDS)	Function Code 4 (NDS to RACF)	Function Code 5 (RACF to KERB)	Function Code 6 (RACF to KERB)
SAF_return_code	Output	Output	Output	Output	Output	Output
RACF_return_code	Output	Output	Output	Output	Output	Output
RACF_reason_code	Output	Output	Output	Output	Output	Output
Function_code	Input	Input	Input	Input	Input	Input
Option_word	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
RACF_userid	Input	Output	Input	Output	Input	Output
Certificate	N/A	Input	N/A	Input	N/A	Input
Application_userid	Output	Input	Output	Input	Output	Input

Usage Notes

1. This service is intended for use by z/OS application servers. It allows them to map between supported application user identities and the corresponding RACF user ID, or to determine the RACF user ID by supplying the corresponding application user identity or digital certificate.
2. An **ALET** must be specified for the SAF_return_code, RACF_return_code, RACF_reason_code parameters, and a single **ALET** specified for all of the remaining parameters.
3. The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, the last word in the parameter list must have a 1 in the high-order bit. If the last word in the parameter list

R_usermap

does not have a 1 in the high-order (sign) bit, the caller receives a parameter list error. The first parameter that can have the high-order bit on, ending the parameter list, is the Application_userid parameter.

4. If the function_code indicates that an application identity is to be returned, and no RACF_userid is supplied, the caller receives a parameter list error.
5. If the function_code indicates that a RACF user ID is to be returned, and no application_userid or certificate is supplied, the caller receives a parameter list error.
6. Specification of an unknown function code, a RACF_userid with a length greater than 8 or an application_userid with a length greater than 246 will result in a parameter list error.
7. If the function_code indicates that a RACF user ID is to be returned, and both an application_userid and a certificate are supplied, the application_userid will be used.
8. If the function_code indicates that an application user identity is to be returned, the caller is expected to supply a 248 byte area for the application_userid parameter. If an application user identity is defined for the RACF_userid specified, R_usermap will update this area with the length and value of the application user identity. This storage must be accessible in the caller's key.
9. If the function_code indicates that a RACF user ID is to be returned, the caller is expected to supply a 9 byte area for the RACF_userid parameter. If a RACF user ID is associated with the application_identity specified, R_usermap will update this area with the length and value of the RACF user ID. This storage must be accessible in the caller's key.
10. The conversion between the supported application user identities (or certificates) and RACF user IDs is dependent on the definition of the application specific segments associated with the RACF USER profile. To convert between Lotus Notes for z/OS user identity and a RACF user ID, the RACF USER profile must have an LNOTES segment containing the SNAME field. To convert between an NDS for z/OS user identity and a RACF user ID, the RACF USER profile must have an NDS segment containing the UNAME field. To convert between a certificate and a RACF user ID, the RACF USER profile must be associated with the certificate.

Note: In the case of Security Server Network Authentication Service identities, local Security Server Network Authentication Service principals are similar to the above: to convert between a Security Server Network Authentication Service local principal identity and a RACF user ID, the RACF USER profile must have a KERB segment containing the KERBNAME field. However, in the case of foreign Security Server Network Authentication Service principals, a KERBLINK class profile must be defined to map the foreign Security Server Network Authentication Service principal to a RACF user identity. The RACF user identity associated with a foreign Security Server Network Authentication Service principal (or multiple foreign Security Server Network Authentication Service principals), does not require a KERB segment.

11. The certificate supplied by the certificate parameter is used only to identify a RACF user ID. It is expected that the certificate was previously verified. Note the following additional details regarding certificate processing:
 - a. All fields as defined for X.509 version 1 certificates must be present and non-null.

- b. X.509 certificates with version numbers greater than 3 are not supported.
 - c. Version 3 certificates with critical extensions are not supported. Noncritical extensions are ignored.
 - d. Subject and issuer names can contain only the following string types:
 - T61STRING - TAG 20
 - PRINTABLESTRING - TAG 19
 - IA5STRING - TAG 22
 - VISIBLESTRING - TAG 26
 - GENERALSTRING - TAG 27
 - e. The length of the serial number plus the length of the issuer's name cannot exceed 245.
 - f. No date validity check is performed on the certificate.
 - g. No signature check is performed on the certificate.
12. If the certificate supplied by the caller is defined to RACF with a status of NOTRUST, R_usermap will return a RACF return code 8, RACF reason code 32, indicating that no user ID is defined to use this certificate.
 13. For function_codes 5 and 6, the application user identity is the Security Server Network Authentication Service principal name. Local principal name is case sensitive, foreign principal is not (the RDEFINE of a KERBLINK profile will fold the name to upper case). R_usermap will accept mixed case input of foreign profile names, but will fold to upper case before attempting to locate the appropriate KERBLINK identity mapping profile. R_usermap output of foreign principal names will always be upper case. Additionally, while local principal names may be supplied fully qualified with the name of the local realm, R_usermap output of local principal names will always be unqualified. Realm qualified names follow a DCE-like convention of /.../realm_name/principal_name.
 14. If the function_code specifies that a RACF user ID is to be returned and the length supplied for the Application_user is greater than the maximum allowed, such as greater than 64 for a Lotus Notes for z/OS user identity, or greater than 240 for Security Server Network Authentication Service user identity, the caller receives the "no mapping between RACF and an application" error.
 15. Check if any logrec entry has been created to ensure R_usermap service was being run successfully and also refer to *z/OS Security Server RACF Diagnosis Guide* for detailed logrec information.

Related Services

R_dceinfo, R_dceruid

R_usermap

Chapter 3. IRRSXT00 Installation Exit

IRRSXT00 installation exit uses the IRRSXT00 module as described in this chapter.

Function

As described in Chapter 1, “Using the RACF Callable Services” on page 1, Linkage Conventions for the Callable Services, IRRSXT00 is invoked by the SAF callable services router before and after RACF is called. It receives as input, a function code indicating which callable service is being called, and the parameter list that will be passed to RACF. The first parameter in the parameter list points to a work area. IRRSXT00 can use the first 152 bytes of this work area. The first word of the work area is set to zero before the pre-RACF call to the exit. IRRSXT00 should set another value in this word to indicate to the post-RACF exit call that it is the second call. The first four words of the work area are passed unchanged from the pre-RACF to the post-RACF exit.

The pre-RACF exit can change the content of the parameter list that will be passed to the external security product. It can also indicate with return codes that the external security product should be bypassed and control returned to the caller. The SAF return code is set based on the exit return code. If the external security product is bypassed, the exit routine must provide all of the output including RACF-compatible return and reason codes that the invokers of the services expect.

The post-RACF exit can look at or change the output from RACF including the RACF return and reason codes. No exit return codes are defined from this exit call. SAF return codes are set based on the RACF return codes, not on an exit return code.

Requirements

Authorization:	*State and key of the user calling the security function
Dispatchable unit mode:	Task of user calling security function
Cross memory mode:	PASN = HASN or PASN not = HASN
AMODE:	Any <i>or</i> 31
RMODE:	Any <i>or</i> 24
ASC mode:	AR mode
Serialization:	Enabled for interrupts
Locks:	No locks held
Control parameters:	The parameter list and the work area are in the primary address space. ALETs are passed for all parameters except the work area. The words containing the ALETs are in the primary address space.

Note: *Most callable services will be supervisor state and key 0. For services that can be invoked unauthorized, it can be problem state and any key.

Interface Registers

Register	AR content	GR content
Input Registers		
0	Any	Function code of service. Refer to the description of IRRPFC in <i>z/OS Security Server RACF Data Areas</i> .
1	0	Address of parameter list
2 - 13	Any	Undefined
14	0	Return address
15	0	Entry point address
Output Registers		
0 - 14	Same as input	Same as input
15	Undefined	Return code

Input

The parameter list is the same as the list that will be passed to the external security product (for example, RACF). The content of the list varies depending on the service requested. See *z/OS Security Server RACF Data Areas* for descriptions of the parameter list, IRRPCOMP, as well as detailed descriptions of structures such as the FSP and CRED, which are passed as parameters on a number of the callable services. See Chapter 2, “Callable Services Descriptions” on page 7 for descriptions of the possible parameter lists.

The first 152 bytes of the work area pointed to by the parameter list can be used by the exit. The rest of the work area is reserved for SAF and RACF. The first four words of the 152-byte area can be used to pass data from the pre-RACF call of the exit to the post-RACF call to the exit. These words are not used by SAF or RACF. The other 136 bytes may be used by RACF services or the SAF router between the calls to the pre-RACF exit and the post-RACF exit.

Output

Returned Data: If the exit indicates that RACF is not to be called, the exit is responsible for setting the RACF return and reason codes and providing any other output data expected by the caller of the requested service.

Refer to Chapter 2, “Callable Services Descriptions” on page 7 for information on which callable services return output to their callers, as well as the format of the output.

Note: The return and reason codes are not stored in the parameter list as they are for SAF exit ICHRTX00. For IRRSXT00, the parameter list contains the addresses of two words in which the return and reason codes must be stored.

The pre-RACF exit routine must restore all registers on return except register 15, which must contain a return code.

The post-RACF exit must also restore all registers except 15. No return codes are defined for the post-RACF exit.

Return Codes: The pre-RACF exit can return one of the following return codes:

Return code	Explanation
0	Exit complete - continue processing and call RACF for further security processing. The exit routine may change the content of the parameter list that will be passed to RACF
200	Exit complete - access authorized. The SAF callable services router sets SAF router return code 0 and returns to the caller of the service, bypassing any further security processing.
204	Exit complete - no decision. The SAF callable services router sets SAF return code 4 and returns to the caller of the service, bypassing any further security processing.
208	Exit complete - access not authorized. The SAF callable services router sets SAF return code 8 and returns to the caller of the service, bypassing any further security processing.
Other	Exit complete - the SAF callable services router sets the SAF return code to the exit-supplied value and returns to the caller of the service, bypassing any further security processing.

Usage Notes

1. IRRSXT00 must be reentrant.
2. IRRSXT00 can receive control in cross memory mode.
3. IRRSXT00 must use BAKR to save registers. No save area is provided on entry. It is called in AR mode and must save both general and access registers.
4. To install the SAF callable services router installation exit, create the load module, name it IRRSXT00, and load it into the link pack area (LPA).
5. IRRSXT00 must provide its own recovery routine. If the exit routine terminates abnormally, its recovery routine gets control. If a recovery routine is not provided or if the recovery routine percolates, the recovery routine of the caller of the service stub will get control.
6. To determine the caller of the SAF callable service, use the function code in register 0 to determine which callable service is being called. Refer to Table 1 on page 7, in Chapter 2, for the list of components and products that are possible callers of each service.

The usage notes that follow the callable service descriptions are also helpful in determining the caller.

Many services receive the CRED as a parameter. The CRED contains an audit function code, defined in macro IRRPAFC, which identifies the z/OS UNIX function calling the service. Additional information on which callable services

IRRSXT00

are called by z/OS UNIX functions can be found in the z/OS Security Server (RACF) Auditor's Guide, under Classes that Control Auditing for z/OS UNIX System Services.

Appendix. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen-readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen-readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Volume I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

This product contains code licensed from RSA Data Security Incorporated.



Programming Interface Information

This manual is intended to describe the RACF callable services. This publication documents intended Programming Interfaces that allow an installation to write programs to obtain the services of RACF.

Trademarks

The following terms are trademarks of IBM Corporation in the United States or other countries or both:

AIX/6000
AnyNet
AT
BookManager
C/370
CICS
CICS/ESA
Current
DB2
DFSMS
DFSMSdfp
DFSMSdss
DFSMShsm
DFSMSrmm
DFSMS/MVS
DFSORT
e-business
eServer

ESCON
GDDM
Hiperbatch
IBM
IBMLink
IMS
Language Environment
Library Reader
MVS/ESA
MVS/SP
MVS/XA
OS/2
OS/390
RACF
Redbooks
Resource Link
RETAIN
RMF
SecureWay
SOMobjects
SystemView
System/390
S/390
VM/ESA
VTAM
Windows
z/OS
z/OS.e
z/VM
zSeries

NetView, Tivoli, TME, and TME 10 are trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the United States, other countries, or both.

Lotus and Lotus Notes are trademarks of Lotus Development Corporation in the United States, or other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

- IRRSIU00 7, 46
- IRRSKA00 7, 9
- IRRSKF00 7, 12
- IRRSKI00 7, 14
- IRRSKO00 7, 20
- IRRSKP00 7, 18
- IRRSMF00 7, 49
- IRRSMI00 7, 52
- IRRSMK00 7, 150
- IRRSMM00 7, 210
- IRRSMR00 7, 55
- IRRSRK00 7, 206
- IRRSPT00 7, 194
- IRRSXP00 7
- IRRSFY00 7, 187
- IRRSQF00 7, 57
- IRRSQS00 7, 59
- IRRSSG00 7, 202
- IRRSSU00 7, 204
- IRRSUD00 7, 134
- IRRSUG00 7, 144
- IRRSUM00 7, 30
- IRRSXT00 219

K

- keyboard 223

L

- licensed documents xv
- LookAt message retrieval tool xv

M

- make IFSP 49
- make IISP 52
- make root IFSP 55
- managed ACEEs 38
- Map application user 212
- mapping
 - get UID-to-user-ID 30
 - GID-to-group-name
 - get 25
- mapping macro
 - BPXYIPCP 6
- message retrieval tool, LookAt xv

N

- NGROUPS_MAX 46, 59, 144
- no timeout ACEEs 38
- notices 225

O

- OCSF Data library 115
- options
 - audit
 - change 108
 - query file security 57
 - query system security 59
- owner
 - change 112

P

- parameter list
 - data field name 61
- parse
 - extract 206
- parse or extract 206
- privilege
 - check 18
- process
 - fork 139
- process owner
 - check 20
- proxy 187
- Ptrace Authority Check 194
- publications
 - on CD-ROM xiv
 - softcopy xiv

Q

- query file security options 57
- query system security options 59

R

- R_admin 61
- R_dceauth 123
- R_dcekey callable service 131
- R_dceruid 134
- R_fork 139
- RACF
 - classroom courses xiv
 - publications
 - on CD-ROM xiv
 - softcopy xiv
- RACF administration
 - classroom courses xiv
- RACF security topics
 - classroom courses xiv
- retrieve or
 - set 150
- retrieve or set Network Authentication Service fields 150
- root IFSP, make 55

S

- S_IRGRP bit, mapping in file mode 4
- S_IROTH bit, mapping in file mode 4
- S_IRUSR bit, mapping in file mode 4
- S_IRWXG bit, mapping in file mode 4
- S_IRWXO bit, mapping in file mode 4
- S_IRWXU bit, mapping in file mode 4
- S_ISGID bit, mapping in file mode 4
- S_ISUID bit, mapping in file mode 4
- S_ISVTX bit, mapping in file mode 4
- S_IWGRP bit, mapping in file mode 4
- S_IWOTH bit, mapping in file mode 4
- S_IWUSR bit, mapping in file mode 4
- S_IXGRP bit, mapping in file mode 4
- S_IXOTH bit, mapping in file mode 4
- S_IXUSR bit, mapping in file mode 4
- saved UIDs/GIDs
 - set 137

- security credentials (CRED)
 - description 2
- security options
 - query file 57
 - query system 59
- security topics for RACF
 - classroom courses xiv
- set
 - effective and saved UIDs/GIDs 137
 - file mode creation mask 210
- set all UIDs 197
- set effective GID/set all GIDs 195
- set effective z/OS UNIX user identifier (UID) 197
- set group name 202
- set ID
 - clear 22
- set supplemental groups 142
- set UID 204
- shortcut keys 223
- supplemental groups
 - get 28, 142
 - set 142
- system security options, query 59

T

- TME
 - administration 61

U

- UID
 - effective
 - set 197
- UID-to-user-ID mapping, get 30
- UIDs
 - all
 - set 197
 - effective
 - set 137
 - get 28
 - saved
 - set 137
- user ID
 - get mapping to UID 30
- USP
 - delete 24
 - initialize 46

W

- work area
 - description 1
- WORK data area
 - description 1

Readers' Comments — We'd Like to Hear from You

z/OS
Security Server RACF
Callable Services

Publication No. SA22-7691-03

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>				

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>				
Complete	<input type="checkbox"/>				
Easy to find	<input type="checkbox"/>				
Easy to understand	<input type="checkbox"/>				
Well organized	<input type="checkbox"/>				
Applicable to your tasks	<input type="checkbox"/>				

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

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

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



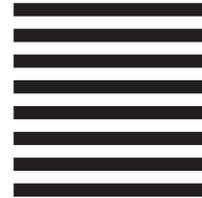
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY
12601-5400



Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5694-A01, 5655-G52

Printed in U.S.A.

SA22-7691-03

