



RACF Parmlib Support - OA52650

Contents

Summary of changes	v	RACPRMCK (Validate parmlib member syntax)	11
Chapter 1. About this document	1	z/OS RACF Messages and Codes	12
Chapter 2. Publications affected	3	IRR messages for RACF database initialization	12
z/OS RACF System Programmer's Guide.	3	RACF database split/merge utility (IRRUT400)	19
IRRPRMxx parmlib member	3	messages	19
Executing IRRUT400.	6	RACPRMCK command messages	20
Example 2a. Splitting a database using the		z/OS RACF Security Administrator's Guide	21
RANGE keyword	10	The RACF command exits	21
Example 2b. Splitting a database	10	Summary of commands and their functions	21
z/OS RACF Callable Services	11	Other authorities	21
R_admin (IRRSEQ00): RACF administration API	11	z/OS MVS Initialization and Tuning Reference	23
z/OS RACF Command Language Reference	11	Overview of IEASYSxx parameters	23
		Overview of parmlib members	23
		IRRPRMxx (RACF parameters).	23

Summary of changes

Table 1. Summary of changes

Version	Date	Nature of Change
Version 1	08/23/17	Initial Version - OA52650 UA93099
Version 2	09/11/17	Updates to the IRRUT400 sections that are affected.

Chapter 1. About this document

The RACF data set name table (ICHRDSNT) and the range table (ICHRRNG) are installation defined load modules that are written, assembled, and link-edited by the system programmer.

This APAR provides the capability for the data set name table (DSNT) and range table to be specified by using a parmlib member, enabling the system programmer to easily update tables as needed. This parmlib member has the prefix of IRRPRM with a two character suffix, which is referenced by the "RACF=" IPL parameter. Having multiple versions of the DSNT and range table allows the system programmer to test different configurations, make modifications or provide backup and recovery if unexpected errors occur.

To use this function, the system programmer must create a parmlib member, IRRPRMxx, that contains data for the data set name table and range table. This information is similar to what can be found in the load modules, ICHRDSNT and ICHRRNG. An IPL is required for a new or updated parmlib to take effect. Specifying the "RACF=" IPL parameter takes precedence over any existing database configuration option.

Chapter 2. Publications affected

z/OS RACF System Programmer's Guide

IRRPRMxx parmlib member

The data set name table (ICHRDSNT) and the database range table (ICHRNG) can be specified by using the parmlib member, IRRPRMxx.

The parmlib member, IRRPRMxx, contains data for the RACF data set name table, range table, and sysplex communications options. One or more parmlib members are parsed and validated for syntax and data validity. If no parmlib members are specified, RACF uses the original method to determine the data set name table and range table.

If you want to specify more than one parmlib members, see “Multiple parmlib members” on page 6.

IRRPRMxx parmlib parameter syntax

The IRRPRMxx parmlib parameter syntax is as follows:

```
DATABASE_OPTIONS
  SYSplex(NOCOMMUNICATIONS | COMMUNICATIONS | DATASHARING)
DATASETNAMETABLE
  ENTRY
    PRIMARYDSN(data-set-name)
    BACKUPDSN(data-set-name)
    UPDATEBACKUP(ALL | NONE | NOSTATS)
    BUFFERS(value)
RANGETABLE
  START(start-value [CHAR | HEX]) ENTRYNUMBER(entry-sequence-number)
```

The IRRPRMxx parmlib parameter descriptions are as follows:

DATABASE_OPTIONS

Header statement for the RACF initialization portion of the parmlib member. This parameter is required for each parmlib member that is processed.

SYSplex

Specifies the sysplex environment that you want.

NOCOMMUNICATIONS | NOCOMM

Specifies that RACF is not enabled for sysplex communications.

COMMUNICATIONS | COMM

Specifies that RACF is enabled for sysplex communications and requests non-data sharing mode at IPL.

DATASHARING

Specifies that RACF is enabled for sysplex communications and requests data sharing mode at IPL.

This parameter is optional. If not specified, SYSplex(NOCOMMUNICATIONS) is the default value.

DATASETNAMETABLE | DSNT

Header statement that marks the beginning of the data set name table. This table can contain multiple entries. The maximum is 90 primary and backup data set pairs.

This parameter is required when defining a data set name table. It is optional if the default data set name table is sufficient. The default data set name table includes a single primary and backup data set pair, for which the operator is prompted. This dataset pair has a default of 255 resident data blocks for buffers and no data duplication to the backup data set.

ENTRY

Header statement that marks the beginning of a single primary and backup data set pair in the data set name table. There can be multiple ENTRY statements to mark each primary and backup data set pair.

This parameter is required for each entry in the data set name table.

PRIMARYDSN(*data-set-name*) | PRIMARY(*data-set-name*) | PRIM(*data-set-name*)

Specifies the name of a primary data set. The data set name must follow data set naming conventions. It can either be a quoted string or an unquoted string. If the data set name is an asterisk, the operator is prompted for a data set name at initialization time.

This parameter is optional. If not specified, the operator is prompted for a primary data set name.

BACKUPDSN(*data-set-name*) | BACKUP(*data-set-name*) | BACK(*data-set-name*)

Specifies the name of a backup data set. The data set name must follow data set naming conventions. It can either be a quoted string or an unquoted string. If the data set name is an asterisk, the operator is prompted for a data set name at initialization time. If the data set name is a blank, the backup is not defined and the operator is not prompted.

This parameter is optional. If not specified, the operator is prompted for a backup data set name.

UPDATEBACKUP | UPDATEBACK

Specifies the update option that you want for the backup data set.

ALL

All updates, including statistics, are to be duplicated on the backup data set.

NONE

No updates are to be duplicated on the backup data set.

NOSTATS

All updates, but no statistics, are to be duplicated on the backup data set.

This parameter is optional. If not specified, UPDATEBACKUP(NONE) is the default value.

BUFFERS(*value*) | BUF(*value*)

Specifies the number of data blocks to be kept resident for the primary data set. If the system is enabled for sysplex communications, 20% of this value is used for the backup data set. The valid values are 0 - 255.

This parameter is optional. If not specified, the default value is 255.

RANGETABLE | RANGE | RNG

Header statement that marks the beginning of the range table. This table can contain multiple entries. The first start entry must be binary zeros and must be arranged in ascending order. The first entry can appear as follows:

```
START('00' HEX) ENTRYNUMBER(1)
```

If only one entry exists in the data set name table, no range table is required.

When more than one entry exists in the range table, each entry must point to an entry in the data set name table. When more than one entry exists in the data set name table, each entry must be referenced by at least one range in the range table; otherwise, an error occurs.

The RANGETABLE parameter is optional. If not specified, a default range table is defined containing a start value of binary zeros that points to the first entry in the data set name table.

START(*start-value* [CHAR | HEX])

Specifies the starting range for a single primary and backup data set pair. The start value can be a string of hexadecimal or alphanumeric characters with or without quotation marks. Quotation marks are required for the following:

- Lowercase characters
- Any blanks included in the string
- A START value that spans multiple lines

If lowercase characters are not enclosed in quotation marks, they are folded to uppercase characters.

The START parameter is optional. If not specified, the default is a single entry of binary zeros pointing to the first entry in the data set name table.

CHAR

Specifies the start value that is interpreted as character data. The maximum length of a *start-value* is 44.

HEX

Specifies the start value that is interpreted as hexadecimal data. The accepted characters are 0 - 9 and uppercase characters A - F. The maximum length of a *start-value* is 88 hexadecimal characters. An even number of characters must be specified.

If not specified, the default is CHAR.

ENTRYNUMBER(*entry-sequence-number*) | ENTRYNUM(*entry-sequence-number*)

Specifies the sequence number (the data set's relative position in the data set name table) of the entry containing the primary and backup data set pair that is associated with the range start value. The valid values are 1 - 90.

The START and ENTRYNUMBER parameters must be specified in pairs.

Syntax rules for IRRPRMxx

The following are syntax rules for the parmlib member, IRRPRMxx:

- Columns 1 - 71 are used for data; Columns 72 - 80 are ignored.
- Comments can appear in columns 1 - 71 and must begin with /* and end with */.

- A section (data set name table, range table or sysplex consideration options) cannot exist multiple times within the same member.
- One or more statements can be entered on a single line or one statement can span multiple lines. For instance, a parameter can appear on one line and its associated value can appear on another. However, the keyword parameter itself cannot span lines. It is recommended that one keyword per line is specified for debugging purposes.
- The continuation of a parameter value is allowed if the parameter value is enclosed in quotation marks (such as the value of the START parameter). There is no explicit continuation character. The quoted string consists of all data up to column 71 and continues in column 1 on the next line.
- At least one blank is required between each parameter, however, not between the parameter and its value. The right parenthesis after the specified parameter value is a sufficient delimiter.
- The parameters and parameter values can be entered in uppercase, lowercase, or in single quotation marks. Any values that are enclosed in single quotation marks are processed as entered. Otherwise, the system converts all characters to uppercase.
- A maximum of 3 parmlib members can be specified on the RACF= parameter during initialization.

Multiple parmlib members

When specifying multiple parmlib members, the data within the data set name table and the range table cannot be split between members. The data set name table and the range table can exist in individual members. However, having some primary and backup data set pairs in one member, and the remaining data set pairs in a different member is not permitted. The same rules apply to the ranges within the range table. The SYSPLEX options can be specified in a separate member.

A maximum of 3 parmlib members can be specified.

If a section (data set name table, range table, or sysplex consideration options) is specified in more than one parmlib member, only the first occurrence is processed. All duplicates are ignored. If a section is specified more than once in a single member, an error message is issued.

If parmlib members are specified, but no data is found in any of the members, an error message is issued. If multiple members are specified and one or more members are empty, the empty members are ignored.

Error processing

If any errors occur during the processing of a parmlib member, an error message is issued to indicate the problem and where it occurred. Parsing continues and additional error messages can be issued as identified.

If errors are detected at the time of IPL, no data from any parmlib member is used. The load modules, ICHRDSNT and ICHRRNG, are not used. The operator is then prompted to specify the name of a single primary and backup data set, or NONE if no primary and backup is to be used. The parmlib members specified at IPL time are documented in the system log along with the data set name where it is found.

Executing IRRUT400

The following job control statements are necessary for executing IRRUT400:

Statement

Use

JOB Initiates the job.

EXEC Specifies the program name (PGM=IRRUT400) or, if the job control statements are in a procedure library, the procedure name. You can also request IRRUT400 processing options by specifying parameters in the PARM field. See “Specifying parameters” on page 8.

STEPLIB DD

Defines the location of an ICHRRNG-style module, which is named by the TABLE keyword, if it does not reside in the system LINKLIB. If used, the data set that is referenced by STEPLIB must be APF-authorized.

SYSPRINT DD

Defines a sequential message data set. The data set can be written to an output device, a tape volume, or a direct access volume.

INDD n DD

Defines a RACF input database. See “Specifying the input database”.

OUTDD n DD

Defines a RACF output database. This statement is not required if you are executing the utility only to identify inconsistencies in a RACF® database, or to unlock a database. See “Specifying the output database”.

Do not allocate this database in the extended addressing area of DASD volumes. To ensure that this database is not allocated in the extended addressing area, the OUTDD n DD statement must not contain the keyword parameter EATTR (unless its value is the default NO). Since EATTR=NO is the default, it is not necessary to include this on the DD statement.

RANGEDD DD

Defines a partitioned data set, or a concatenation of partitioned data sets, in which to locate the member that is specified by the RANGE parameter. If RANGEDD is not defined, IRRUT400 searches the current system parmlib instead. If the RANGE parameter is not specified, RANGEDD is ignored.

If you are redistributing the profiles across more than one data set, you must also provide a *range table*. The range table indicates which profiles are placed in each output data set (using the TABLE or RANGE keyword to make the determination). See “The database range table” and “Selecting the output data set.” If any of the input data sets are RACF-protected, you must have at least UPDATE authority for those data sets.

Selecting the output data set

If multiple output data sets are created, the utility uses the range table to determine which profiles to copy to which output data sets. Therefore, you must supply range definitions that indicate the range of profiles to be placed on each data set.

You can specify the name of a partitioned data set member containing the RACF parmlib range definitions with the RANGE parameter in the EXEC statement. For more information, see “Specifying parameters” on page 8. For information about the correct member syntax, see “Syntax rules for IRRPRMxx” on page 5. IRRUT400 does not use any of the data set names that are defined in the parmlib member. You must ensure that the range definitions correspond correctly to the output data sets that are specified by OUTDD n in the JCL.

Alternatively, you can specify the name of a module that contains an ICHRRNG-style range table in the TABLE parameter in the EXEC statement. See "Specifying parameters." "The database range table" describes the format of the ICHRRNG range table.

Before restarting your system to use a new split RACF database, ensure that the data set name table is consistent with the range table that is to be used for that system. The tables can be specified using a RACF parmlib member or using the ICHRDSNT and ICHRRNG load modules. Do not rely on IRRUT400 to validate all of the definitions in a RACF parmlib member. The RACPRMCK command can be used to validate the RACF parmlib member.

Specifying parameters

You can specify a number of parameters in the PARM field of the EXEC statement of the step executing IRRUT400. The syntax for the parameters is similar to that of the TSO command language. They can be separated by one or more blanks. Embedded blanks are not allowed. Any keyword can be abbreviated to the number of initial characters that uniquely identify that keyword. The specification of redundant or contradictory keywords is considered an error.

LOCKINPUT/NOLOCKINPUT/UNLOCKINPUT

You must specify one of these keywords.

RACF sysplex data sharing: If your system is running in read-only mode, you cannot specify LOCKINPUT or UNLOCKINPUT for IRRUT400.

LOCKINPUT does not allow updates to be made to the specified input data sets, even after the utility terminates. Statistics are updated, however.

If the RACF database is *locked*, a user attempts to logon, and RACF must update the user's profile, the logon might be allowed, or it might fail. It fails if:

- This is the user's first logon of the day and RACF is not in data sharing mode.
- The password is being changed.
- The user is entering the correct password after previously entering an incorrect password.

Otherwise, because RACF is only making a statistical update to the profile, the logon is allowed.

LOCKINPUT locks only the input data sets; it does not lock the output data sets.

Attention:

- When using LOCKINPUT against an active database, do not schedule maintenance spanning midnight. If the RACF database remains locked past midnight when RACF is not in data sharing mode, users will be unable to submit new jobs or log on, unless you disable the gathering of logon statistics by issuing a SETROPTS NOINITSTATS command. All steps that require a locked database must be performed on the same calendar day.
- When you are using LOCKINPUT and running IRRUT400, any activity updating the RACF database will fail with either an ABEND483 RC50 or ABEND485 RC50.

NOLOCKINPUT does not change the status of the data sets, nor does it prevent updates to the input data sets. NOLOCKINPUT is intended to be used for completely inactive RACF databases. If you use it for active RACF

databases, all systems sharing the database should have nothing running, such as users logging on, which could result in a write to the active database.

If NOLOCKINPUT is specified and updates occur to the input data sets, the results of the utility and the content of the output data sets will be unpredictable.

UNLOCKINPUT can be used to unlock all data sets that were previously locked by LOCKINPUT. This re-enables your input data set and allows it to be updated.

In most cases, you probably do not need to unlock your input data sets. After using IRRUT400 to create one or more new output data sets, you probably want to use the new output data sets, not the old input data sets. The output data sets are not locked by LOCKINPUT. If, for some reason, the utility is unable to create a valid output data set, it unlocks the input data sets for you. You might need to use UNLOCKINPUT if you mistakenly lock the wrong data set, or if you change your mind after locking a data set.

RANGE(member)/TABLE(table-name)/NOTABLE

Use the RANGE or TABLE keyword to indicate the name ranges for separating profiles into multiple output data sets. NOTABLE is the default and implies that all profiles are to be copied into a single output data set, which is identified by OUTDD1.

The RANGE keyword enables you to specify the name of a RACF parmlib member that defines the ranges for selecting an output data set for each profile. IRRUT400 looks for the specified member in the current system parmlib concatenation unless RANGEDD is included in your JCL to indicate a different partitioned data set or concatenation. For more information about the correct member syntax, see "Syntax rules for IRRPRMxx" on page 5. Syntax errors that are found in the specified member cause IRRUT400 to terminate, even if the errors reside outside of the RANGETABLE section. The use of system symbols in the range table is not recommended, since symbol values for the environment in which you run IRRUT400 can differ from those in another system configuration.

The TABLE keyword enables you to specify the name of an ICHRRNG-style module that defines ranges for selecting an output data set for each profile. While the TABLE value can be any member name, the member format must be that of an ICHRRNG module as described in "The database range table". IRRUT400 loads the specified module from the system LINKLIB. You can override the location by defining a STEPLIB in your JCL. The data set that is referenced by STEPLIB must be APF-authorized.

If you are using the utility to merge or split a database into multiple output data sets, you must specify either RANGE or TABLE to indicate in which data set to place the profiles. You must ensure that the range definitions correspond correctly to the output data sets that are specified by OUTDD n in the JCL. IRRUT400 does not use the data set name table that is defined in a RACF parmlib member or an ICHRDSNT module. It is the system programmer's responsibility to ensure that the set of data sets that are specified by OUTDD n is consistent with the data set name table you intend to use for your RACF database.

FREESPACE(percent)/NOFREESPACE

This keyword allows you to control the amount of free space left in index blocks created for the output data sets. You can specify that from 0 to 50 percent of the space within the index block is to be left free. The sequence set (level one) will contain the specified percentage of free space; level two will

contain one seventh of the specified percentage. Index levels higher than two will contain approximately seven percent free space.

NOFREESPACE [equivalent to FREESPACE(0)] is the default.

The amount of free space you specify should depend on the frequency of updates to the RACF database. For normal RACF database activity, a value of 30 is suggested. If frequent database updates occur, use more.

Note that this keyword does not determine the amount of free space in the database; it affects only the index blocks.

ALIGN/NOALIGN

This keyword allows you to control profile space allocation. Specifying ALIGN forces segments that occupy multiple 256-byte slots to be placed so that they do not span 4096-byte physical blocks. Having a single physical block can decrease the I/O needed to process these segments. Specifying NOALIGN (the default) causes no special alignment.

DUPDATASETS/NODUPDATASETS

This keyword allows you to control the processing of DATASET entries with identical names from different input data sets. Specifying DUPDATASETS indicates that duplicates are allowed and that all DATASET entries are to be processed. If you specify NODUPDATASETS and the utility encounters duplicate entries on different data sets, the utility copies the DATASET entry from the input data set identified by the lowest-numbered ddname. When NODUPDATASETS is in effect, duplicates occurring on a single input data set are all accepted, assuming that they do not conflict with an entry from another data set earlier in the selection sequence. NODUPDATASETS is the default.

Example 2a. Splitting a database using the RANGE keyword

In this example, IRRUT400 splits a RACF database containing a single data set into two output data sets. IRRUT400 assigns profiles to the output data sets using range table definitions in a member named IRRPRM02 that does not reside in the current parmlib concatenation. The member can be found in INSTALL.WORK.PARMLIB as indicated by RANGEDD.

```
//J2      JOB
//        EXEC  PGM=IRRUT400,PARM='NOLOCKINPUT,RANGE(IRRPRM02) '
//SYSPRINT DD  SYSOUT=A
//INDD1   DD   DSN=SYS1.RACF,DISP=OLD
//OUTDD1  DD   DSN=SYS2.RACF1,DISP=(,KEEP),
//          UNIT=SYSDA,VOL=SER=VOL1,
//          DCB=DSORG=PSU,
//          SPACE=(CYL,5,,CONTIG)
//OUTDD2  DD   DSN=SYS2.RACF2,DISP=(,KEEP),
//          UNIT=SYSDA,VOL=SER=VOL2,
//          DCB=DSORG=PSU,
//          SPACE=(CYL,20,,CONTIG)
//RANGEDD DD   DSN=INSTALL.WORK.PARMLIB,DISP=SHR
```

Example 2b. Splitting a database

In this example, IRRUT400 splits a RACF database containing a single data set into three output data sets. IRRUT400 assigns profiles to the output data sets using an ICHRRNG-style range table in a module named RNGNEW. The load module resides in library INSTALL.LINKLIB, which is APF-authorized.

```
//J2      JOB
//        EXEC  PGM=IRRUT400,PARM='NOLOCKINPUT, TABLE(RNGNEW) '
//SYSPRINT DD  SYSOUT=A
//INDD1   DD   DSN=SYS1.RACF,DISP=OLD
//OUTDD1  DD   DSN=SYS2.RACF1,DISP=(,KEEP),
```



```

//          UNIT=SYSDA,VOL=SER=VOL1,
//          DCB=DSORG=PSU,
//          SPACE=(CYL,5,,CONTIG)
//OUTDD2   DD  DSN=SYS2.RACF2,DISP=(,KEEP),
//          UNIT=SYSDA,VOL=SER=VOL2,
//          DCB=DSORG=PSU,
//          SPACE=(CYL,20,,CONTIG)
//OUTDD3   DD  DSN=SYS2.RACF3,DISP=(,KEEP),
//          UNIT=SYSDA,VOL=SER=VOL3,
//          DCB=DSORG=PSU,
//          SPACE=(CYL,5,,CONTIG)
//STEPLIB  DD  DSN=INSTALL.LINKLIB,DISP=SHR

```

z/OS RACF Callable Services

R_admin (IRRSEQ00): RACF administration API

Usage notes

4. 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)
- RACPRMCK

z/OS RACF Command Language Reference

RACPRMCK (Validate parmlib member syntax)

Purpose

Use the RACPRMCK command to validate the syntax of one or more RACF parmlib members and verify that the content of the member is valid prior to IPLing.

The input to the command is the name of one or more parmlib members. The output of the command, consists of one or more messages that are displayed to indicate whether the RACF parmlib member is valid or not.

Issuing options

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

As a RACF TSO command?	As a RACF operator command?	With command direction?	With automatic command direction?	From the RACF parameter library?
Yes	Yes	No	No	No

For information on issuing this command as a RACF TSO command, refer to RACF TSO commands.

For information on issuing this command as a RACF operator command, refer to RACF operator commands.

Authorization required

When issuing this command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. For details about OPERCMDS resources, see Controlling the use of operator commands in *z/OS Security Server RACF Security Administrator's Guide*.

Syntax

For the key to the symbols used in the command syntax diagrams, see Syntax of RACF commands and operands. The complete syntax of the RACPRMCK command is:

```
RACPRMCK
  MEMBER(member-name1 [, member-name2 [, member-name3]])
```

Parameters

MEMBER(member-name1 [, member-name2 [, member-name3]])

Specifies the name of one or more RACF parmlib members that you want to validate. The member name must reside in a data set that is part of the parmlib concatenation. A maximum of 3 parmlib members can be specified.

z/OS RACF Messages and Codes

IRR messages for RACF database initialization

**IRRY101E RACF DETECED AN ERROR WHILE PARSING PARMLIB.
TOO MANY SUFFIXES WERE SPECIFIED ON THE RACF= PARAMETER.**

Explanation: There are too many suffixes specified on the RACF= parameter during initialization. The maximum number of suffixes that is allowed is 3.

System action: RACF no longer uses parmlib for initialization. A default data set name table and range table is used. The message ICH502I is issued to prompt the operator for a single primary/backup data set pair. A default value of 255 is used for the number of resident data blocks assigned to the primary data set, data is not duplicated to the backup data set, and RACF is not enabled for sysplex communications. The range table contains one entry with a start value of binary zeros.

Operator response: Contact the system programmer.

System programmer response: Correct the number of suffixes specified on the RACF= parameter.

Routing code: 2

Descriptor code: 4

**IRRY102E RACF DETECTED AN ERROR WHILE PARSING PAMRLIB.
ALLOCATION FAILED FOR THE LOGICAL PARMLIB DATA SET
CONCATENATION. IEFPRMLB RETURN CODE *return-code*
REASON CODE *reason-code***

Explanation: An error occurred attempting to allocate the logical parmlib data set concatenation using the IEFPRMLB service. The return code from the allocate request is indicated by *return-code* and the reason code is indicated by *reason-code*. The return and reason codes are in hexadecimal format.

System action: If this message is issued during RACF initialization, RACF no longer uses parmlib to initialize. A default data set name table and range table is used. The message ICH502I is issued to prompt the operator for a single primary/backup data set pair. A default value of 255 is used for the number of resident data blocks assigned to the primary data set, data is not duplicated to the backup data set, and RACF is not enabled for sysplex communications. The range table contains one entry with a start value of binary zeros.

If you are issuing the RACPRMCK command to validate parmlib members, the command processing continues to the next member if specified.

If you are running the IRRUT400 utility, processing stops.

Operator response: Contact the system programmer.

System programmer response: Refer to the IEFPRMLB macro documentation in *z/OS MVS Programming: Authorized Assembler Services Guide* for an explanation of the return code and reason code that is indicated in the message. Look for any additional messages that were generated during IEFPRMLB processing.

Routing code: 1

Descriptor code: 2

**IRRY103E RACF DETECED AN ERROR WHILE PARSING PARMLIB.
LOAD FAILED FOR MODULE IEEMB878
ABEND CODE *abend-code* REASON CODE *reason-code***

Explanation: An unexpected error occurred while attempting to load module IEEMB878 during parmlib processing. The error is indicated by *abend-code* and *reason-code*. The abend and reason codes are in hexadecimal format.

System action: If this message is issued during RACF initialization, RACF no longer uses parmlib to initialize. A default data set name table and range table is used. The message ICH502I is issued to prompt the operator for a single primary and backup data set pair. A default value of 255 is used for the number of resident data blocks assigned to the primary data set, data is not duplicated to the backup data set, and RACF is not enabled for sysplex communications. The range table contains one entry with a start value of binary zeros.

If you are issuing the RACPRMCK command to validate parmlib members, the command processing continues to the next member if specified.

If you are running the IRRUT400 utility, processing stops.

Operator response: Contact the system programmer.

System programmer response: Refer to *z/OS MVS System Codes* for the indicated abend code and reason code. Look for any additional messages that were generated during IEEMB878 processing.

Routing code: 1

Descriptor code: 2

**IRRY104E RACF DETECTED AN ERROR WHILE PARSING PARMLIB.
ERROR READING PARMLIB MEMBER *member-name*
IEEMB878 RETURN CODE *return-code***

Explanation: Module IEEMB878 returned an unexpected return code while reading the indicated parmlib member. The return code is in hexadecimal format.

System action: If this message is issued during RACF initialization, RACF no longer uses parmlib to initialize. A default data set name table and range table is used. The message ICH502I is issued to prompt the operator for a single primary and backup data set pair. A default value of 255 is used for the number of resident data blocks assigned to the primary data set, data is not duplicated to the backup data set, and RACF is not enabled for sysplex communications. The range table contains one entry with a start value of binary zeros.

If you are issuing the RACPRMCK command to validate parmlib members, the command processing continues to the next member if specified.

If you are running the IRRUT400 utility, processing stops.

Operator response: Contact the system programmer.

System programmer response: Verify that all data sets associated with the logical parmlib concatenation are

IRRY105E • IRRY106E

available and the member is located within the concatenation. For the IRRUT400 utility, if the RANGEDD card is specified, verify that all data sets associated with the RANGEDD card are available and that the member is located within those data sets.

Problem Determination: Return code indicated by *return-code* from module IEEMB878 identifies the error as follows:

Table 2. Return code descriptions

Return code	Description of error
08	I/O error detected.
0C	OPEN of parmlib failed.
10	Member not found.
14	Data not valid in parameter list.
18	Cannot access data set.
1C	Conversion error

Routing code: 1

Descriptor code: 2

**IRRY105E RACF DETECTED AN ERROR WHILE PARSING PARMLIB.
ERROR FOUND IN MEMBER *member-name*
LINE *xxxx*: *keyword-1* WAS EXPECTED BEFORE *keyword-2***

Explanation: An error was found during syntax processing of the indicated parmlib member. The keyword indicated by *keyword-1* was expected before *keyword-2*, but was either missing, not where it was expected, or typed incorrectly.

System action: If this message is issued during RACF initialization, RACF no longer uses parmlib to initialize. A default data set name table and range table is used. The message ICH502I is issued to prompt the operator for a single primary and backup data set pair. A default value of 255 is used for the number of resident data blocks assigned to the primary data set, data is not duplicated to the backup data set, and RACF is not enabled for sysplex communications. The range table contains one entry with a start value of binary zeros.

If you are issuing the RACPRMCK command to validate parmlib members or running the IRRUT400 utility, processing continues for any additional syntax or validation errors. The command or utility completes with a failure.

Operator response: Contact the system programmer.

System programmer response: Edit the indicated parmlib member and add or correct the keyword indicated by *keyword-1* where the error was found. Refer to the *z/OS Security Server RACF System Programmer's Guide* for the correct syntax. Consider running the RACF TSO command, RACPRMCK, to verify that all errors are removed prior to IPLing.

Routing code: 2

Descriptor code: 4

**IRRY106E RACF DETECTED AN ERROR WHILE PARSING PARMLIB.
ERROR FOUND IN MEMBER *member-name*
LINE *xxxx*: *keyword-1* WAS SEEN IN COLUMN *yyyy*
EXPECTED: *expected-keyword-list***

Explanation: An error was found during syntax processing of the indicated parmlib member. Either a keyword was missing, not where it was expected, or typed incorrectly.

System action: If this message is issued during RACF initialization, RACF no longer uses parmlib to initialize. A default data set name table and range table is used. The message ICH502I is issued to prompt the operator for a single primary and backup data set pair. A default value of 255 is used for the number of resident data blocks assigned to the primary data set, data is not duplicated to the backup data set, and RACF is not enabled for sysplex communications. The range table contains one entry with a start value of binary zeros.

If you are issuing the RACPRMCK command to validate parmlib members or running the IRRUT400 utility,

processing continues for any additional syntax or validation errors. The command or utility completes with a failure.

Operator response: Contact the system programmer.

System programmer response: Edit the indicated parmlib member and correct the keyword indicated by *keyword-1*. Choose from the list of expected keywords that are found in the error message to use in place of *keyword-1*. Refer to *z/OS Security Server RACF System Programmer's Guide* for the correct syntax. Consider running the RACF TSO command RACPRMCK to verify that all errors are removed prior to IPLing.

Routing code: 2

Descriptor code: 4

**IRRY107E RACF DETECTED AN ERROR WHILE PARSING PARMLIB.
ERROR FOUND IN MEMBER *member-name*
LINE *xxxx*: *keyword* KEYWORD VALUE IS NOT VALID.**

Explanation: The value associated with the indicated keyword is not valid.

System action: If this message is issued during RACF initialization, RACF no longer uses parmlib to initialize. A default data set name table and range table is used. The message ICH502I is issued to prompt the operator for a single primary and backup data set pair. A default value of 255 is used for the number of resident data blocks assigned to the primary data set, data is not duplicated to the backup data set, and RACF is not enabled for sysplex communications. The range table contains one entry with a start value of binary zeros.

If you are issuing the RACPRMCK command to validate parmlib members or running the IRRUT400 utility, processing continues for any additional syntax or validation errors. The command or utility completes with a failure.

Operator response: Contact the system programmer.

System programmer response: Edit the indicated parmlib member and correct the keyword value that is associated with *keyword*. For more information, see *z/OS Security Server RACF System Programmer's Guide*

Consider running the RACF TSO command, RACPRMCK, to verify that all errors are removed prior to IPLing.

Routing code: 2

Descriptor code: 4

**IRRY108E RACF DETECTED AN ERROR WHILE PARSING PARMLIB.
ERROR FOUND IN MEMBER *member-name*
LINE *xxxx*: {PRIMARYDSN | BACKUPDSN} DATA SET NAME IS NOT VALID.**

Explanation: The string that is associated with the PRIMARYDSN or BACKUPDSN keyword does not follow data set naming conventions.

System action: If this message is issued during RACF initialization, RACF no longer uses parmlib to initialize. A default data set name table and range table is used. The message ICH502I is issued to prompt the operator for a single primary and backup data set pair. A default value of 255 is used for the number of resident data blocks assigned to the primary data set, data is not duplicated to the backup data set, and RACF is not enabled for sysplex communications. The range table contains one entry with a start value of binary zeros.

If you are issuing the RACPRMCK command to validate parmlib members or running the IRRUT400 utility, processing continues for any additional syntax or validation errors. The command or utility completes with a failure.

Operator response: Contact the system programmer.

System programmer response: Edit the indicated parmlib member and correct the value that is associated with the PRIMARYDSN or BACKUPDSN keyword to conform to data set naming conventions. Consider running the RACF TSO command, RACPRMCK, to verify that all errors are removed prior to IPLing.

Routing code: 2

Descriptor code: 4

**IRRY109E RACF DETECTED AN ERROR WHILE PARSING PARMLIB.
ERROR FOUND IN MEMBER *member-name*.
THE FIRST RANGE TABLE START VALUE MUST BE BINARY ZEROS.**

Explanation: The first START keyword in the range table is not binary zeros.

System action: If this message is issued during RACF initialization, RACF no longer uses parmlib to initialize. A default data set name table and range table is used. The message ICH502I is issued to prompt the operator for a single primary and backup data set pair. A default value of 255 is used for the number of resident data blocks assigned to the primary data set, data is not duplicated to the backup data set, and RACF is not enabled for sysplex communications. The range table contains one entry with a start value of binary zeros.

If you are issuing the RACPRMCK command to validate parmlib members or running the IRRUT400 utility, processing continues for any additional syntax or validation errors. The command or utility completes with a failure.

Operator response: Contact the system programmer.

System programmer response: Edit the indicated parmlib member and correct the value that is associated with the first START keyword. The first START keyword can be specified as follows:

START('00' HEX)

Consider running the RACF TSO command, RACPRMCK, to verify that all errors are removed prior to IPLing.

Routing code: 2

Descriptor code: 4

**IRRY110E RACF DETECTED AN ERROR WHILE PARSING PARMLIB.
ERROR FOUND IN MEMBER *member-name*.
START VALUE FOUND ON LINE *xxxx* CAUSES THE RANGE TABLE ENTRIES TO BE
OUT OF ORDER.**

Explanation: The value associated with the START keyword on the indicated line number causes the range table to be out of order. All START keyword values must appear in ascending order.

System action: If this message is issued during RACF initialization, RACF no longer uses parmlib to initialize. A default data set name table and range table is used. The message ICH502I is issued to prompt the operator for a single primary and backup data set pair. A default value of 255 is used for the number of resident data blocks assigned to the primary data set, data is not duplicated to the backup data set, and RACF is not enabled for sysplex communications. The range table contains one entry with a start value of binary zeros.

If you are issuing the RACPRMCK command to validate parmlib members or running the IRRUT400 utility, processing continues for any additional syntax or validation errors. The command or utility completes with a failure.

Operator response: Contact the system programmer.

System programmer response: Edit the indicated parmlib member and correct the order of the entries so that all START keyword values appear in ascending order. Consider running the RACF TSO command, RACPRMCK, to verify that all errors are removed prior to IPLing.

Routing code: 2

Descriptor code: 4

**IRRY111E RACF DETECTED AN ERROR WHILE PARSING PARMLIB.
RANGE TABLE ENTRY WITH START VALUE
start-value
HAS AN ENTRYNUMBER THAT IS NOT VALID.**

Explanation: The range table entry with the indicated start value does not have a matching entry in the data set name table. The value specified with the keyword ENTRYNUMBER is too large.

System action: If this message is issued during RACF initialization, RACF no longer uses parmlib to initialize. A default data set name table and range table is used. The message ICH502I is issued to prompt the operator for a single primary and backup data set pair. A default value of 255 is used for the number of resident data blocks assigned to the primary data set, data is not duplicated to the backup data set, and RACF is not enabled for sysplex

communications. The range table contains one entry with a start value of binary zeros.

If you are issuing the RACPRMCK command to validate parmlib members, the command continues for any additional syntax or validation errors and completes with a failure.

Operator response: Contact the system programmer.

System programmer response: Check the parmlib members that were specified on the RACF= parameter during initialization or as input to the RACF TSO command, RACPRMCK, for a mismatch between the data set name table and range table. The value specified with the keyword ENTRYNUMBER could be too large. Consider running the RACF TSO command, RACPRMCK, to verify that all errors are removed prior to IPLing.

Routing code: 2

Descriptor code: 4

**IRRY112E RACF DETECTED AN ERROR WHILE PARSING PARMLIB.
DSNT ENTRYNUMBER *entry-number* IS NOT ASSOCIATED
WITH ANY ENTRY IN THE RANGE TABLE.**

Explanation: There are no entries in the range table that would allow access to the data set name table entry indicated by *entry-number*. The *entry-number* indicates the data set's relative position in the data set name table.

System action: If this message is issued during RACF initialization, RACF no longer uses parmlib to initialize. A default data set name table and range table is used. The message ICH502I is issued to prompt the operator for a single primary and backup data set pair. A default value of 255 is used for the number of resident data blocks assigned to the primary data set, data is not duplicated to the backup data set, and RACF is not enabled for sysplex communications. The range table contains one entry with a start value of binary zeros.

If you are issuing the RACPRMCK command to validate parmlib members, the command continues to process for any additional syntax or validation errors and completes with a failure.

Operator response: Contact the system programmer.

System programmer response: Check the parmlib members that were specified on the RACF= parameter during initialization or as input to the RACF TSO command, RACPRMCK, for a mismatch between the data set name table and range table. Consider running the RACF TSO command, RACPRMCK, to verify that all errors are removed prior to IPLing.

Routing code: 2

Descriptor code: 4

**IRRY113I DATA SET NAME TABLE DATA WAS NOT FOUND IN PARMLIB.
DEFAULT VALUES WILL BE USED.**

Explanation: No data was found for the data set name table in any of the specified parmlib members but data was found for a range table.

System action: If this message is issued during RACF initialization, RACF no longer uses parmlib to initialize. A default data set name table and range table is used. The message ICH502I is issued to prompt the operator for a single primary and backup data set pair. A default value of 255 is used for the number of resident data blocks assigned to the primary data set and data is not duplicated to the backup data set. RACF continues to use parmlib for the range table and the sysplex communication options, if they exist.

If you are issuing the RACPRMCK command to validate parmlib members, the command continues to process for any additional syntax or validation errors and completes with a failure.

Operator response: Contact the system programmer to ensure that this was the intended system action.

System programmer response: No response needed if this was intentional, otherwise edit a parmlib member to include data for a data set name table. Refer to *z/OS Security Server RACF System Programmer's Guide* for the correct syntax.

Routing code: 2

Descriptor code: 4

**IRRY114E RACF DETECTED AN ERROR WHILE PARSING PARMLIB.
NO DATA FOUND IN PARMLIB MEMBER(S).**

Explanation: Parmlib members were specified on the RACF= parameter for initialization but no data was found in any member.

System action: If this message is issued during RACF initialization, RACF no longer uses parmlib to initialize. A default data set name table and range table is used. The message ICH502I is issued to prompt the operator for a single primary and backup data set pair. A default value of 255 is used for the number of resident data blocks assigned to the primary data set, data is not duplicated to the backup data set, and RACF is not enabled for sysplex communications. The range table contains one entry with a start value of binary zeros.

If you are issuing the RACPRMCK command to validate parmlib members or running the IRRUT400 utility, processing terminates.

Operator response: Contact the system programmer.

System programmer response: Check the system log to verify that the correct parmlib members were specified. Verify that the specified members contain the RACF configuration data. Refer to the *z/OS Security Server RACF System Programmer's Guide* for the correct syntax.

Routing code: 2

Descriptor code: 4

IRRY115I RACF IS NOT USING PARMLIB FOR INITIALIZATION.

Explanation: One or more errors occurred while processing parmlib members for RACF initialization.

System action: A default data set name table and range table is used. The message ICH502I is issued to prompt the operator for a single primary and backup data set pair. The range table will contain one entry with a start value of binary zeros.

Operator response: Contact the system programmer.

System programmer response: Check the system log for any additional error messages that were issued for RACF initialization during parmlib processing. Correct these errors and refer to *z/OS Security Server RACF Messages and Codes* for explanation of these messages as needed. Issue the RACF TSO command, RACPRMCK, to verify that all errors are removed prior to IPLing.

Routing code: 1

Descriptor code: 2

**IRRY116E RACF DETECTED AN ERROR WHILE PARSING PARMLIB.
ERROR FOUND IN MEMBER *member-name*.
THE NUMBER OF ENTRIES IN THE DATA NAMES TABLE
EXCEEDS THE MAXIMUM OF 90.**

Explanation: The data set name table contains more entries than allowed. The maximum number that is allowed is 90.

System action: If this message is issued during RACF initialization, RACF no longer uses parmlib to initialize. A default data set name table and range table is used. The message ICH502I is issued to prompt the operator for a single primary/backup data set pair. A default value of 255 is used for the number of resident data blocks assigned to the primary data set, data is not duplicated to the backup data set, and RACF is not enabled for sysplex communications. The range table contains one entry with a start value of binary zeros.

If you are issuing the RACPRMCK command to validate parmlib members, the command continues for any additional syntax or validation errors and completes with a failure.

Operator response: Contact the system programmer.

System programmer response: Edit the indicated parmlib member and correct the number of entries in the data set names table. Consider running the RACF TSO command, RACPRMCK, to verify that all errors are removed prior to IPLing.

Routing code: 2

Descriptor code: 4

IRRY1171 NO VALID SUFFIXES WERE SPECIFIED ON THE RACF= PARAMETER.

Explanation: The RACF= parameter was specified during initialization but without any valid suffixes.

System action: The RACF= parameter is ignored. A default data set name table and range table are used. The message ICH502I is issued to prompt the operator for a single primary/backup data set pair. A default value of 255 is used for the number of resident data blocks assigned to the primary data set, data is not duplicated to the backup data set, and RACF is not enabled for sysplex communications. The range table contains one entry with a start value of binary zeros.

Operator response: Contact the system programmer.

System programmer response: Either specify a valid PARMLB member suffix on the RACF= parameter or remove the RACF= parameter.

Routing code: 2

Descriptor code: 4

IRRY118E UNABLE TO ESTABLISH ESTATE ENVIRONMENT. PROCESSING
system-action

Explanation: An ESTAE recovery environment cannot be established.

System action: Processing is indicated by *system-action*. The *system-action* is either CONTINUES WITHOUT RECOVERY, which is issued during RACF initialization, or TERMINATED, which is issued during other non-initialization type functions.

Operator response: Contact the system programmer.

System programmer response: Ensure that RACF and the operating system are properly installed. If they are, report this message, including the message ID, to your IBM support center.

Routing code: 2

Descriptor code: 4

IRRY119E *abend-code[-yy]* ABEND WHILE PROCESSING LOAD MODULE IRRPRM00.

Explanation: An abend occurred while running the IRRPRM00 load module. If a reason code was specified with *abend-code*, it is displayed as *yy*. The returned abend code and reason code is in hex.

System action: Processing continues with recovery. If this message is issued during RACF initialization, RACF becomes inactive. If you are issuing the RACPRMCK command to validate parmlib members or running the IRRUT400 utility, processing terminates.

Operator response: Contact the system programmer.

System programmer response: Examine the console log and the system dump for more information pertaining to the abend. For an explanation of the abend code, see *z/OS MVS System Codes*.

Routing code: 2

Descriptor code: 4

RACF database split/merge utility (IRRUT400) messages

| **IRR65042I Specified RANGE member *member-name* found in dataset *dataset-name* on volume *volume-name* .**

| **Explanation:** The member that is specified for the RANGE keyword is located in the indicated data set and volume. If the member does not contain RACF parmlib range definitions or contains syntax errors, you can expect a subsequent error message.

| **System action:** Utility processing continues.

| **System Programmer Response:** None.

| **IRR65043I** Error encountered processing specified RANGE member *member-name*. Processing terminated.

| **Explanation:** The utility encountered an error locating, reading, or parsing the member that is specified for the RANGE keyword. The associated messages, with a prefix of IRRY, are issued to provide additional information.

| **System action:** Utility processing stops.

| **System Programmer Response:** For information about the syntax rules for RACF parmlib range definitions, see "IRRRPMxx parmlib member" on page 3 in *z/OS Security Server RACF System Programmer's Guide*. Correct any syntax errors that are found in the specified member, even if found outside the RANGETABLE section. IRRUT400 only performs partial checking of statements outside the RANGETABLE section. For complete validation, use the RACPRMCK command documented in *z/OS Security Server RACF Command Language Reference*.

| **IRR65044I** No ranges found in specified RANGE member *member-name*. Processing terminated.

| **Explanation:** The utility was unable to identify any RACF parmlib range definitions in the member that is specified for RANGE keyword.

| **System action:** Utility processing stops.

| **System Programmer Response:** For information about syntax rules for RACF parmlib range definitions, see "IRRRPMxx parmlib member" on page 3 in *z/OS Security Server RACF System Programmer's Guide*.

RACPRMCK command messages

IRRY301I No errors found in PARMLIB member(s).

Explanation: The RACPRMCK command is issued to validate the parmlib members. The command completed successfully with no errors found.

System action: None.

User response: No response required.

IRRY302I Errors found in PARMLIB member(s).

Explanation: The RACPRMCK command is issued to validate the parmlib members. The command completed with one or more error messages issued. A list of errors follow this message indicating the errors that are found. If more than 50 errors are encountered, only the first 50 messages are displayed.

System action: Command completes processing.

User response: Correct the errors found in the parmlib members and re-issue the RACPRMCK command. For more information about the correct syntax of the parmlib member, see *z/OS Security Server RACF System Programmer's Guide*.

IRRY303I Too many members were specified with the MEMBER keyword.

Explanation: There are too many members specified with the MEMBER keyword. The maximum number of members allowed is 3.

System action: Command processing ends.

User response: Correct the number of members specified with the MEMBER keyword.

IRRY304I *keyword* keyword was not specified with the RACPRMCK command.

Explanation: The RACPRMCK command was issued without the *keyword* keyword. This keyword is required.

System action: Command processing ends.

User response: Re-issue the RACPRMCK command with the required keyword.

IRRY305I The *command* command is not allowed to be issued.

Explanation: The *command* command was issued as an operator command, but did not have the authority for one of the following reasons:

- There is no RACF defined user ID associated with the operator.
- A profile in the OPERCMDS class is preventing the command from being issued by the user ID.

System action: Command processing ends.

User response: If the command was issued from an operator console, ensure that the console is logged on. If the command was issued through some other means, ensure that the command is issued from a RACF defined user ID. If the message, ICH408I, was issued before this message, an OPERCMDS profile is preventing access. Contact your RACF security administrator to get access to the OPERCMDS profile.

Administrator response: If appropriate, permit the user ID to the OPERCMDS profile that is protecting the command in question.

z/OS RACF Security Administrator's Guide

The RACF command exits

RACF provides a command exit (IRREVX01) that is invoked before and after the execution of all RACF TSO commands except the block update command (BLKUPD), RACDCERT, RACLINK, RACMAP, RACPRIV, RACPRMCK, RVARY, and RACF operator commands, such as RESTART, TARGET, and SIGNOFF.

This exit permits the installation to perform functions that include:

- Checking additional authorization
- Modifying authorization checking when these commands are issued
- Changing the options specified on the command
- Stopping the command from executing

There are other command exits that are invoked during processing of certain commands. See *z/OS Security Server RACF System Programmer's Guide* for information on which exits are invoked for each command.

Summary of commands and their functions

RACF commands allow you to list, modify, add, and delete profiles for users, groups, connect entries, and resources. Table 3 shows, in alphabetic order, each of the commands and its functions.

Table 3. Functions of RACF commands

RACF command	Command functions
RACPRMCK	<ul style="list-style-type: none"> • Validate the syntax of one or more RACF parmlib members. • Verify that the data within the RACF parmlib members is valid for the data set name table and range table.

Other authorities

Table 4 on page 22 shows the commands and operands you can issue for reasons not already covered previously.

Table 4. Commands and operands you can issue for miscellaneous reasons

User ID relationship	Commands and operands you can issue if you have this authority
User ID is current user	<p>ALTUSER only with NAME or DFLTGRP</p> <p>LISTUSER only with user ID</p> <p>PASSWORD or PHRASE only with PASSWORD or INTERVAL</p>
User ID is high-level qualifier of data set name (or qualifier supplied by a command installation exit)	<p>ADDSD with all operands</p> <p>ALTDSD with all operands except OWNER or GLOBALAUDIT</p> <p>DELDSD with all operands</p> <p>LISTDSD with all operands</p> <p>PERMIT with all operands</p> <p>SEARCH with all operands</p>
None	<p>RVARY¹ with all operands</p>
None	<p>RACF MVS™ Operator Commands:</p> <p>DISPLAY Authority granted by OPERCMDS class: See RACF operator command profiles: Naming conventions and z/OS Security Server RACF Command Language Reference.</p> <p>RACPRMCK Authority granted by OPERCMDS class</p> <p>RESTART Authority granted by OPERCMDS class</p> <p>SET Authority granted by OPERCMDS class</p> <p>SIGNOFF Authority granted by OPERCMDS class: See RACF operator command profiles: Naming conventions and z/OS Security Server RACF Command Language Reference.</p> <p>STOP Authority granted by OPERCMDS class</p> <p>TARGET Authority granted by OPERCMDS class</p> <p>Any RACF TSO command issued as an operator command</p>

Table 4. Commands and operands you can issue for miscellaneous reasons (continued)

User ID relationship	Commands and operands you can issue if you have this authority
¹	Although no special authority is needed to issue this command, the system operator must supply the appropriate RVARYPW password, as established by the SETROPTS command with the RVARYPW operand, to approve any change in RACF status.

z/OS MVS Initialization and Tuning Reference

Overview of IEASYSxx parameters

Table 5 summarizes all system parameters that can be placed in an IEASYSxx or IEASYS00 member (or specified by the operator). Detailed discussions of these parameters are provided in later sections of the IEASYSxx topic.

Note: PAGE and GRS are the only mandatory parameters that have no default. They must be specified.

The GRSRNL parameter is mandatory when the GRS= parameter is specified as JOIN, TRYJOIN, START, or STAR. The GRSRNL parameter is ignored when GRS=NONE.

Table 5. Overview of IEASYSxx parameters

Parameter	Use of the parameter
RACF	Specifies the parmlib member or members (IRRPRMxx) that are used to locate statements that define the RACF database configuration.

Overview of parmlib members

Table 6. Characteristics of parmlib members

Supplied by IBM®	Required or optional	Affects performance directly?	Read at IPL or at command?	Allows listing of parameters at IPL or command?	Response to errors (N/A = Not applicable)			System symbols support?	Concatenated parmlib support?
					Syntax error	Read error	Unsupported parameters		
IRRPRMxx: Defines parameters for RACF.									
Yes, in SYS1.SAMPLIB	Optional	No	IPL	No	Operator responds to prompt. System programmer is required to specify primary and backup data sets.	Same as with syntax error.	Same as with syntax error.	Yes	Yes

IRRPRMxx (RACF parameters)

The IRRPRMxx parmlib member contains data for the RACF data set name table, range table, and sysplex communications options. Use an IRRPRMxx member to specify statements that describe the RACF data set name table and the range table, that were previously defined by ICHRDSNT and ICHRRNG.

For the data set name table, the statements include the primary and backup data set name, update backup options, and resident index blocks.

Sysplex communication options can also be specified in the parmlib member.

Each individual section can be in its own parmlib member, however, no section can be split between multiple members. A maximum of 3 parmlib members can be specified.

For complete information about the IRRPRMxx parmlib member, including syntax and parameters, see Specifying RACF database options in *z/OS Security Server RACF System Programmer's Guide*.

IBM-supplied sample for IRRPRMxx

IBM provides the sample, IRRPRMXX, found in SYS1.SAMPLIB.



Printed in USA