



RTE-6/VM Quick Reference Guide

**Software Technology Division
11000 Wolfe Road
Cupertino, California 95014-9804**

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THE MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Hewlett-Packard Company.

Printing History

The Printing History below identifies the edition of this manual and any updates that are included. Periodically, update packages are distributed which contain replacement pages to be merged into the manual, including an updated copy of this printing history page. Also, the update may contain write-in instructions.

Each reprinting of this manual will incorporate all past updates; however, no new information will be added. Thus, the reprinted copy will be identical in content to prior printings of the same edition with its user-inserted update information. New editions of this manual will contain new information, as well as all updates.

To determine what manual edition and update is compatible with your current software revision code, refer to the Manual Numbering File or the Computer User's Documentation Index. (The Manual Numbering File is included with your software. It consists of an "M" followed by a five digit product number.)

Second Edition	Dec 1983	CI File System added
Third Edition	May 1988	Software Rev. 5.0 (Rev. 5000)
Fourth Edition	May 1989	Software Rev. 5.1 (Rev. 5010)
Update 1	Oct 1990	Software Rev. 5.2 (Rev. 5020)
Fifth Edition	Jun 1993	Software Rev. 6.0 (Rev. 6000)

Conventions

Conventions Used in this Manual	1-1
File Descriptor Convention	1-1
Terms Used in this Manual	1-6
Boot Procedure	1-7

Conventions Used in this Manual

This manual uses the following conventions to describe command syntax:

CAPITAL LETTERS	Commands or parameters that must be entered exactly as shown are in capital letters; however, CI always accepts lowercase input.
[]	Optional parameters are in brackets. If additional parameters follow an omitted parameter, commas must be used as placeholders.
<>	Parameters enclosed in angle brackets are optional in some cases and required in others.
	Parameter choices are separated by vertical bars.
,	Delimiters between commands and parameters are commas or spaces.
lowercase letters	Terms representing actual parameters (variables) are in lowercase letters.
single underline	Single underlined parameters have values returned by the system.

File Descriptor Convention

A file descriptor has up to 63 characters and one of the following formats, depending upon the application (the first two are equivalent):

CI FILE FORMAT:

```
/directory/subdirectory/filename::type:size:recordlength>dspart
```

COMBINED FORMAT:

```
subdirectory/filename::directory:type:size:recordlength>dspart
```

FMGR FORMAT:

```
filename:securitycode:cartridge:type:size:recordlength>dspart
```

A file name has up to 16 characters, plus a type extension of up to four additional characters and generally cannot include the characters “at sign” (@), minus (-), slash (/), period (.), colon (:), greater than (>), comma (,), space (), or left bracket ([). All alphabetical characters are treated as upper case.

A type extension is separated from the file name by a period (.) and has up to four characters. The standard type extensions are:

.cmd	CI command file
.dat	data file
.dbg	symbolic debug file
.dir	directory or subdirectory entry
.doc	document file
.err	error message file
.ftn	FORTRAN source file
.ftni	FORTRAN source include file
.hlp	Help file
.lib	library of relocatables
.lod	LINK command file
.lst	compiler listing
.mac	Macro source file
.maci	Macro source include file
.map	load map listing
.mlb	Macro library file
.mnf	manual numbering file
.mrg	merge control file
.pas	Pascal source file
.pasi	Pascal source include file
.rel	relocatable (binary) file
.run	program file
.snp	system snapshot file
.sys	system file
.txt	text file

File type is specified by an integer in the range zero to 32767. Default is type 3. Standard file types are:

0	I/O device.
1	Random-access file; 128-word records.
2	Random-access file; user-specified record length.
3 & 4	Sequential-access text file; variable-length records.
5	Relocatable object code file.
6	Executable program file.
7	Absolute binary file.

File size is specified by an integer in the range -32768 to 32767 , inclusive. A positive number allocates space in blocks (128 words each); a negative number allocates space in 128-block sections.

Record length (in words) must be specified for a type 2 file. For other file types, this field is ignored.

dspart is one of:

>nodeid[logon/password]
or
>[logon/password]nodeid

where:

nodeid is either the node number or the name and <logon> is a user logon for that node. If the <nodeid> is missing, the current node is used. If a user <logon> is missing the default ds user logon is used.

Note: Brackets ([]) in the above examples are literally required as shown if <logon> is used.

FMGR Format. A subset of file descriptors, used with the FMGR file system, use the namr format.

namr = name:security:cartridge:file type:file size:record size
or
namr = logical unit number

where:

security	<0	Write and read protected
	0	Not protected (default)
	>0	Write protected
cartridge	<0	LU number
	0	First available cartridge (default)
	>0	FMGR cartridge reference number
file type	0	Non-disk file
	1	128-word record length, random access
	2	User selected record length, random access
	3	(and greater) variable record length, sequential access
	4	Source program
	5	Relocatable program
	6	RTE load module
	7	Absolute program
	>7	User defined
file size		Specified in blocks (2 sectors = 1 block = 128 words)
	+n	= allocate n blocks
	-n	= allocate n 128 block multiples
	-1	= allocate remaining space on cartridge

record size
Used only when file is type 2

Masking Format. An extension to the basic file descriptor that specifies a range of files.

Its form is: maskname.maskqualifier

maskname The file name in the format described above with the addition of wildcard characters (@) or (-).

One example of a maskname is:

/TEACHER/CLASSES/@ATTENDANCE.TXT.c83.

Notice the two periods: one to introduce the extension and one to introduce the qualifier.

Masking characters for file names and extensions are:

@ meaning zero or one or more characters.

- meaning any single character, except a blank.

maskqualifier

The mask qualifier is separated from the fileDescriptor with a period. Recognized forms are:

(User.group)

Mask by specified user (see User masking, below).

a Access time stamp mask (see Time stamps, below).

b Only files that have not been backed up.

c Create time stamp mask (see Time stamps, below).

d Include all files in a directory if the directory name matches.

e Everywhere; search all mounted volumes, including FMGR cartridges.

k Same as S except cancels D and enables full match levels.

m Return extent entries on FMGR directories.

n No directories; opposite of the D qualifier.

o Open files only.

p Purged files only.

s Search all new file directories (excludes FMGR cartridges).

t Temporary files only.

u Update time stamp mask (see Time stamps, below).

w Walk (not run) through FMGR directories (see below).

- x Only hierarchical files with extents.
- y Return correct extent information on directories (requires an additional disk access for each directory).

Time stamps and ranges are legal in qualifiers. The form here is:

[a|c|u] [firstTime] [-][endTime]

a	Accessed
c	Created
u	Updated

firstTime and endTime are specified in timestamp format: YYMMDD.HHMMSS and can be truncated on the right for the resolution desired. firstTime defaults to time zero (January 1, 1970) and endTime defaults to present time. If no endTime or range dash is given, firstTime means an absolute time such as 83015 (April 15, 1983), 8304 (April 1983), or 83 (1983). Years from 00 to 37 mean 2000 to 2037 (time stamps can only record dates through 2037). Directory files are not masked by time stamps as their times never change.

User masking is accomplished by specifying the user name as follows:

(user.group) returns only files belonging to the given user.

(@.group) returns only those files belonging to users in the specified group.

(user.@) or (user) returns only those files belonging to the given user in the GENERAL group, which is implied.

(user) or (@.) returns files that have no group specified. (These files were created with pre 5.0 software.)

Walk through FMGR cartridges. FMGR directories are written on the disk in a staggered fashion. The order of access to files in these directories corresponds to the order of the directories containing them. This is known as walking. An application for which the order of access to files is not important can gain in performance by using a faster, non-staggered manner of file access known as running. The masking routines use the fast way unless the W qualifier is set or the buffer area supplied is 8K words, in which case no speed is gained by running.

K, D, and S options. K is useful in CO commands where the directory structure is to be copied but only files matching the mask are wanted. D is set by CO and causes all files in matching directories to be copied. K will override the D, while retaining the directory structure. S does not override the D. If neither D nor K is present, the directory structure is lost.

Terms Used in this Manual

The following definitions apply throughout this manual. Terms not described below are defined under each command description.

prog	One- to five-character program name. Examples: A, PROGA, TIMER, LRUN.
lu	Logical unit number, in the range 0 to 255, inclusive. Refers to a physical I/O device. LU 1 always refers to the user terminal. LU 0 refers to “the bit bucket,” a nonexistent device (unwanted data can be sent to LU 0).
file	File descriptor, which unambiguously specifies a single file.
mask	A masking file descriptor (usually containing wildcard characters) that specifies a range of files.
file lu	Either a file descriptor or a logical unit can be specified.
mask lu	Either a mask or a logical unit can be specified.
parm	One parameter is allowed.
parm*n	As many as <i>n</i> parameters are allowed. In most applications, unspecified parameters default to zero or zero-length strings.
prog file	Either a program name or a file descriptor can be specified. Refer to the RU command for details.

Boot Procedure

1. Select the S-Register for display on the computer front panel.
2. Press CLEAR DISPLAY.
3. Set the S-Register bits as follows:

Bits: Enter:

- | | |
|-------|---|
| 0-2 | Surface number of the disk where the RTE-6/VM system subchannel starts, or, for CS/80 drives, the system drive unit number. |
| 3-4 | 0 (reserved). |
| 5 | 0 for standard boot-up. |
| 6-11 | Octal select code of the disk. |
| 12 | 1 to indicate a manual boot from the S-Register. |
| 13 | 0 (reserved). |
| 14-15 | Loader ROM selection (number of the ROM cell containing the Disk Boot Loader). |
4. Press STORE.
 5. Press PRESET, IBL, and PRESET (again) to load contents of the Disk Loader ROM.
 6. Press RUN.

CI Commands

AG (Modify Partition Priority Aging)*	2-1
AS (Assign Partition)*	2-1
ASK (Display a Prompt and Read a Response)	2-1
BL (Examine or Modify Buffer Limits)*	2-2
BR (Break Program Execution)*	2-2
CD (Change Working Directory)	2-2
CL (List Mounted Disks)	2-3
CN (Control Device)	2-3
CO (Copy files)	2-4
CR (Create File)	2-4
CRDIR (Create Directory/Subdirectory)	2-5
CU (CPU Utilization)*	2-5
DC (Dismount Disk Volume)	2-5
DL (Directory List)	2-6
DN (Down a Device or I/O Controller)*	2-7
ECHO (Display Parameters to Terminal)	2-7
EQ (Display I/O Controller Status)*	2-7
EQ (Buffering)*	2-8
EX (Exit)	2-8
GO (Resume Suspended Program)*	2-8
HE (Help)*	2-9
IF-THEN-ELSE-FI (Control Structure)	2-9
IN (Initialize Disk Volume)	2-9
IS (Compare Strings or Numbers)	2-10
IT (Interval Timer)*	2-11
LI (List Files)	2-12
LU (Display/Modify Device Assignment)*	2-14
MC (Mount Disk Volume)	2-15
MO (Move Files)	2-15
OF (Stop or Remove Program)*	2-15
ON (Schedule Program)*	2-16
OWNER (Display/Change Owner)	2-16
PATH (Display or Modify UDSP)	2-17
POLL (Polling Function)	2-17
PR (Change Program Priority)*	2-18
PROT (Display/Change Protection)	2-18
PU (Purge Files)	2-18
PWD (Display Working Directory)	2-18

QU (Timeslice Quantum)*	2-19
RETURN (Return from Command File)	2-19
RN (Rename File, Directory or Subdirectory)	2-20
RP (Restore Program File)	2-20
RS (Restart Session Progenitor)*	2-21
RU (Run Program)*	2-21
SET (Display or Define Variables)	2-22
SL (Display/Modify Session LU Information)*	2-22
SS (Suspend Program)*	2-23
ST (Display Program Status)*	2-23
SZ (Display or Modify Program Size)*	2-23
TI (Display Time)*	2-24
TM (Display or Set System Clock)	2-24
TO (Display or Set Device Timeout)*	2-24
TR (Transfer to Command File)	2-25
UL (Unlock Shareable EMA Partition)*	2-25
UNPU (Unpurge Files)	2-25
UNSET (Delete User-Defined Variable)	2-26
UP (Up a Device)*	2-26
UR (Release Reserved Partition)*	2-26
VS (Display/Change VMA Program Size)*	2-26
WD (Display/Change Working Directory)	2-27
WH (System Status Reporting)*	2-27
WHILE-DO-DONE (Control Structure)	2-28
WHOSD (Report User of Directory or Volume)	2-28
WS (Display or Modify VMA Working Set Size)*	2-28
XQ (Run Program without Wait)	2-29
?/HELP(Help)	2-29
Command Stack Editor	2-30
\$1 – \$9 Positional Variables	2-30
Predefined Variables	2-31
User-Defined Variables	2-32

AG (Modify Partition Priority Aging)*

Purpose: Modify the rate a partition's priority number is increased and turn on or off partition priority aging.

Syntax: AG numb|of

numb Number of 10-millisecond intervals to be used as the aging rate, in range 10 to 32767.

of Turns off partition priority aging.

AS (Assign Partition)*

Purpose: Assign a reserved partition to a program.

Syntax: AS prog partnum

prog Program name, up to five characters.

partnum Partition number. Partnum = 0 removes the current assignment.

ASK (Display a Prompt and Read a Response)

Purpose: Display a question or prompt, read the response from the terminal, and pass it back to the scheduling program in Return_S.

Syntax: ASK 'character string'

character string

Any question or prompt the user desires. The string must be enclosed in backquotes ("").

* This command can be entered in response to a BREAKMODE or CM prompt.

BL (Examine or Modify Buffer Limits)*

Purpose: Allow the general user to examine the current buffer limits and the System Manager to change the current buffer limits.

Syntax: BL [lowerlimit [upperlimit]]

lowerlimit Lower limit specified in number of words. Used by the System Manager only. If upper limit is changed and lower limit is not specified, it defaults to 1.

upperlimit Upper limit specified in number of words. Used by the System Manager only. If lower limit is changed and upper limit is not specified, it remains the same as the existing upper buffer limits.

BR (Break Program Execution)*

Purpose: Set a flag to allow limited communication with a program.

Syntax: BR prog

prog Program name, up to five characters.

CD (Change Working Directory)

Purpose: Changes the working directory.

Syntax: CD [- |directory]
CD old new

- Change current directory to the previous directory (\$OLDPWD).

directory Change current directory to *directory*. The default for *directory* is the value of the \$HOME variable.

old new Substitute the string *new* for the string *old* in the current working directory name, \$WD, and change to this new directory.

* This command can be entered in response to a BREAKMODE or CM prompt.

CL (List Mounted Disks)

Purpose: Display all mounted disk volumes.

Syntax: CL

CN (Control Device)

Purpose: Control peripheral devices.

Syntax: CN lu function [parm*4]

lu Logical Unit of device to receive the control request.
function The control function code (0-63B) as defined in the function field of CNTWD (listed in the *RTE-6/VM Programmer's Reference Manual*), or one of the following mnemonic codes.

Mnemonic Code	Equiv. Octal Function Code	Parm 1-4 Definition	Action
TO	11B	# lines/ page	Issue top-of-form or line spacing on printer
RW	4	None	Rewind cassette tape
EO	1	None	Write end-of-file
FF	13B	None	Forward space file
BF	14B	None	Backward space file
FR	3	None	Forward space record
BR	3	None	Backward space record
None	0	None	Clear device

For magnetic tapes and cassette tapes, the function parameter defaults to rewind tape, for printers, form feed.

parm*4 Optional parameters that specify additional device details as appropriate for a given driver. Specific meanings for each parameter may be found in the *RTE-6/VM Programmer's Reference Manual*.

CO (Copy files)

Purpose: Copy one or more files between directories and/or I/O devices.

Syntax: CO mask|lu1 mask|lu2 [parm]

mask|lu1 Source file or device.

mask|lu2 Destination file or output device.

parm One of the following characters (default is A):

A ASCII records; no checksum.

B Binary absolute; checksum performed.

C Clear backup bit on source after copying.

D Overwrite duplicate files.

N No carriage control on source.

P Purge source after copying.

Q Quiet. No access time recorded on source.

T Truncate destination to length of valid data.

U Replace duplicates if update time is older.

CR (Create File)

Purpose: Create a disk file.

Syntax: CR file

file File descriptor (up to 63 characters) in one of the following formats:

STANDARD:

/directory/subdirectory/filename:::type:size:recL length

COMBINED:

subdirectory/filename:::directory:type:size:recL length

FMGR:

filename:sc:crn:type:size:recLength

Refer to Chapter 1 for more details on file descriptors.

CRDIR (Create Directory/Subdirectory)

<i>Purpose:</i>	Create a global directory or a subdirectory.
<i>Syntax:</i>	CRDIR directory [lu]
directory	Directory name (up to 63 characters). The name can include an optional size subparameter specified in number of blocks with the format: directory:::size Default size is the track size of the disk, typically 48 or 64 blocks for a hard disk and 30 or 16 for a flexible disk. Directory size is extended as needed.
lu	LU of volume on which to create global directory. Default is LU of working directory. Ignored for a subdirectory.

CU (CPU Utilization)*

<i>Purpose:</i>	Display a bar graph of CPU display registers showing the percentage of CPU utilization.
<i>Syntax:</i>	CU on off
on	Turn display on.
off	Turn display off.

DC (Dismount Disk Volume)

<i>Purpose:</i>	Dismount a disk volume.
<i>Syntax:</i>	DC lu
lu	LU number of the disk volume to be dismounted. Must be a positive number.

* This command can be entered in response to a BREAKMODE or CM prompt.

DL (Directory List)

<i>Purpose:</i>	List files in a directory.
<i>Syntax:</i>	DL [mask [options [file lu [msc]]]]
mask	Specifies files to be displayed. Default is all files in the working directory.
options	Information to be shown for displayed files (can be listed without delimiters in any order). <ul style="list-style-type: none">A ACCESS time.B Indicate files that have not been BACKED UP with an asterisk (*).C CREATION time.E Type EXTENSION (for sorting only).F FILE type.L File LOCATION (block address on disk).M MAIN file size in blocks, excluding extents.N NUMBER of records.O Display OPEN files.P File PROTECTION in the form owner [/group]/other.R Length (in words) of longest RECORD.S File SIZE in blocks, including extents.T Indicate TEMPORARY files.U UPDATE time.W Number of WORDS in file, up to EOF.X Indicate files with EXTENTS.Y Security code (FMGR files only).* Options F, W, N, S, X, and P.! All options.+ Ascending sort by item specified.- Descending sort by item specified.
file lu	File or LU where the DL output is to be stored.
msc	Master security code for the system. Required when Y or ! options are specified.

DN (Down a Device or I/O Controller)*

Purpose: Declare a device or I/O controller down, that is, unavailable for use by the RTE system.

Syntax: DN,,lu or DN,eqt

lu System LU of the device to be declared down.

eqt Equipment Table (EQT) entry number of the I/O controller to be declared down.

ECHO (Display Parameters to Terminal)

Purpose: Display parameters, separated by commas, at the terminal.

Syntax: ECHO [parameters]

parameters

One or more parameters separated by blanks or commas. Positional, user-defined, and predefined variables can be included in the string. If the parameter is omitted, a blank line is displayed.

EQ (Display I/O Controller Status)*

Purpose: Display description and status of an I/O controller, as recorded in the Equipment Table (EQT) entry.

Syntax: EQ eqt

eqt EQT entry number of an I/O controller.

* This command can be entered in response to a BREAKMODE or CM prompt.

EQ (Buffering)*

Purpose: Change the automatic buffering designation for a particular I/O controller.

Syntax: EQ eqt un|bu

eqt Equipment Table (EQT) entry number of an I/O controller.

un Turn off (unbuffer) buffering.

bu Turn on buffering.

EX (Exit)

Purpose: Terminate CI and print the message “Finished”.

Syntax: EX

GO (Resume Suspended Program)*

Purpose: Resume execution of a suspended program.

Syntax: GO [prog [parm*5]]

prog Name of the suspended program.

parm*5 Parameters to be passed to the program (only if the program has suspended itself).

* This command can be entered in response to a BREAKMODE or CM prompt.

HE (Help)*

Purpose: Provide explanation of an error and guidance in possible corrective action.

Syntax: HE [keyword [lu]]

keyword A select group of eight or less characters identifying the error for which an explanation is requested. All keywords and the corresponding explanations are contained in a disk-resident HELP file. Default is the last error that occurred in that session.

lu LU of the device where the explanation is to be sent. Default is the session user's terminal.

IF-THEN-ELSE-FI (Control Structure)

Purpose: Allows decision making in a command file.

Syntax: IF command-list1
THEN command-list2
[ELSE command-list3]
FI

command-listn

A list of commands either one command per line or multiple commands per line separated by semicolons. A command-list can be null.

IN (Initialize Disk Volume)

Purpose: Prepare a blank disk volume for system use.

Syntax: IN lu [blocks [OK]]

lu LU number of the disk volume to be initialized.

blocks Number of blocks to be reserved at the beginning of the disk LU for the boot extension and diagnostics (default is zero).

OK Suppress user prompt.

* This command can be entered in response to a BREAKMODE prompt.

IS (Compare Strings or Numbers)

Purpose: Compare two character strings or numbers.

Syntax: IS string1 operator string2 [option]

string1 A numeric or character string.

operator Relational operator indicating the relation being tested. The two sets of operators recognized are as follows:

=	or	EQ	Equal to
<>	or	NE	Not equal to
<	or	LE	Less than
<=	or	LE	Less than or equal to
>	or	GT	Greater than
>=	or	GT	Greater than or equal to

string2 A numeric or character string.

option Specifies special comparison instructions. The possible values are as follows:

- i Integer comparison. A suffix of B following string1 or string2 in either uppercase or lowercase indicates an octal value. A leading minus (-) sign is accepted for decimal values.
- a Do not casefold alphabetic characters before comparison.

IT (Interval Timer)*

Purpose: Set execution time and interval of repetition when a program is scheduled with the ON command. Place a program into the time list.

Syntax: IT program [res [mpt [hr [min [sec [ms]]]]]]]

To take a program out of the time list:

IT program

program Name of the program to be placed in the time list.

res Time interval resolution:

1 tens of milliseconds

2 seconds

3 minutes

4 hours

mpt Multiplier used in conjunction with time interval resolution value. Can be in the range of 0 to 4095. If 0 is specified, the program runs only once.

hr min sec ms

Optional parameters setting the initial time in terms of hour, minute, second, and tens of milliseconds. Default for any parameters is zero (0).

* This command can be entered in response to a BREAKMODE or CM prompt.

LI (List Files)

Purpose: List files to terminal.

Syntax: LI [options] filemask

filemask File descriptor mask for files to be listed.

options Precede options with a dash. Group them together, as in `-nhx`, or separately, as in `-n -h -x`; `-s 10 -e 15` is equivalent to `-se 10 15`.

- `-a` List ASCII text (default for file types 0, 3, and 4).
- `-w` List octal words (default for file types other than file type 0, 3, and 4).
- `-o` List octal bytes.
- `-i` List signed integer words.
- `-b` List binary words.
- `-h` List hexadecimal words.
- `-d` List ASCII text with Display Functions around special characters.
- `-n` List line/record numbers.
- `-s ln` Set listing to start at *ln*.
- `-e ln` Set listing to end at *ln*.
- `-x` Quit listing at EOF and do not prompt.
- `-$` Always prompt at end of file.
- `-m` Fold long lines; treat lines longer than 79 characters as multiple lines for pagination.
- `-t` Truncate trailing blanks on text listings.
- `-f` Force type 1 access, list blocks in octal words.
- `-c` File has carriage-control characters in column 1.
- `-q` Quiet file access, do not record access time.
- `-l fl` Divert listing to file *fl*. Precede *fl* with tilde (`~`) to overlay existing file or with plus sign (`+`) to append to existing file.
- `-y` Yes, list each file matching `<filemask>`.
- `-r rsz`
Set maximum record size if more than 512 characters is desired.
- `-p pgsz`
Set lines per page to *pgsz* (1..32767); do not paginate if *pgsz* is zero.
- `-> /cmds/`
Execute initial command string *cmds* at start of file. Use back quotes (```) around string entered from CI.

-p /str/

Redefine 'More...' prompt. Delimit *str* with character other than space or comma. Use back quotes (‘) around string.

The following substitutions occur within *str*:

%f	file name.
%l	current line number.
%p	percentage through file as in default.
%w	window or page number viewed.
%%	%.

The following listing commands can be entered at the 'More...' or 'End...' prompts; *n* is a number (from 1 to 2147483647) that can precede the listing command:

space or l	List next page or next <i>n</i> lines.
return	List rest of file or goto line <i>n</i> .
a or q	Abort list; 'a' moves to next masked file, 'q' quits entire listing.
+	List next line or skip forward <i>n</i> lines.
-	Skip backward 1 line or <i>n</i> lines from top line in window.
b	Skip backward 1 page or <i>n</i> pages from top line in window.
g or .	Goto line <i>n</i> ; 'g' to list page, '.' to list 1 line.
\$	List last window.
%	Goto line <i>n</i> percent through file.
'rex'	Search for next occurrence of regular expression, <i>rex</i> from current window or line <i>n</i> . Null string searches for the last string entered. In interactive mode, eliminate trailing apostrophe and terminate with <return>.
f/rex/	Same as 'rex' except with user-defined delimiters. In interactive mode, use <return> instead of delimiters.
'rex'	Search backward for regular expression <i>rex</i> from current window or line <i>n</i> . Null string searches for last string entered. In interactive mode, eliminate trailing backquote and terminate with <return>.
@rex	Show all lines containing pattern from current window or from line <i>n</i> .
km	Mark top window line or line <i>n</i> with character <i>m</i> , which must be an alphabetical character ('A..'Z').
:m	Go to line marked with character <i>m</i> ; list <i>n</i> lines.
um	List until line marked with <i>m</i> ; list no more than <i>n</i> lines.

p	Set page size to <i>n</i> and list a page.
oc	Toggle or reset runstring options, including listing modes (a, b, h, i, o, or w).
s	Set starting file line to window top. LI never starts before this line.
e	Set ending file line to window bottom. LI never advances past this line.
#f	Move to file # <i>n</i> or prompt for file number <i>f</i> . (Requires VMA).
#+[<i>i</i>]	Move forward <i>i</i> selected files or 1 file. (Requires VMA).
#-[<i>i</i>]	Move backward <i>i</i> selected files or 1 file. (Requires VMA).
#?[<i>f</i>]	Show window of selected files around current file or of files starting at <i>f</i> . (Requires VMA).
=	Display file name, current line.
nfile	Add new file name, <i>file</i> , to list of files to be displayed and move to this file. (Requires VMA).
r	Purge file listed.
? or h	Help.
z	Suspend LI; restart with system GO command.

LU (Display/Modify Device Assignment)*

Purpose: Display information associated with a device specified by its LU number. Selected STATUS can be modified by the System Manager.

Syntax: LU lu [eqt [subchannel]]

lu System LU for which information or reassignment is desired.

eqt Used by the System Manager only. Assigns the Equipment Table (EQT) entry number to the LU specified. If 0 is specified, the LU becomes the bit bucket.

subchannel Used by the System Manager only. Assigns subchannel number (0 to 63) to specified LU.

* This command can be entered in response to a BREAKMODE or CM prompt.

MC (Mount Disk Volume)

<i>Purpose:</i>	Mount a disk volume and make its contents available.
<i>Syntax:</i>	MC lu
lu	LU number of the disk volume to be mounted. Must be a positive number.

MO (Move Files)

<i>Purpose:</i>	Move files from one directory to another, within a disk volume.
<i>Syntax:</i>	MO mask1 mask2
mask1	Source file.
mask2	Destination file.

OF (Stop or Remove Program)*

<i>Purpose:</i>	Stop a scheduled program or release a program ID segment.
<i>Syntax:</i>	OF [prog [parm]]
prog	Program name.
parm	An optional parameter specifying action to be taken. Possible values are: 0 – remove from time list (default). 1 – terminate immediately; release disk tracks. 8 – terminate immediately and remove ID segment. ID – same as 8.

* This command can be entered in response to a BREAKMODE or CM prompt.

ON (Schedule Program)*

Purpose: Schedule a program for execution. Up to five parameters and the command string can be passed to the program.

Syntax: ON [program [NO [parm*5]]]

program Name of a program to be scheduled.

NO(W) A program that is normally scheduled by the system clock is scheduled immediately. It may be entered as NOW.

parm*5 Up to five parameters can be passed to the program when it is scheduled.

OWNER (Display/Change Owner)

Purpose: Display or change the owner of a CI volume, directory, or a subdirectory.

Syntax: OWNER directory|luV [newowner]

directory Name of directory; no wildcard characters are allowed.

luV A CI volume number followed by a 'V'.

newowner Name of new owner; if omitted, the name of the owner is displayed.

* This command can be entered in response to a BREAKMODE or CM prompt.

PATH (Display or Modify UDSP)

- Purpose:* Display or modify a User-Definable Directory Search Path (UDSP).
- Syntax:* PATH [-E]
PATH [-E] [-N:n] udspnum [dirname1
[dirname2 [... [dirnamen]...]]]
PATH [-E] -F file|lu
- E Turn off echo; non-error messages are not displayed.
- N:n Display or modify the specified entry. Set *n* equal to 1 for UDSP #0 (home directory); otherwise, set *n* to a value between 1 and the USDP depth.
- udspnum Specify the UDSP number. The values for udspnum are as follows:
O Home directory.
n UDSP number between one and the number of UDSPs defined for this session (maximum is eight).
-A All UDSPs defined for current session.
- dirname Specify the directory name. The following special characters can be used:
. Use the working directory that is current when the UDSP is referenced.
! Delete this UDSP or entry; this character must be the only dirname in the command line.
- F file|lu Indicate that commands will be from the specified file or LU.

POLL (Polling Function)

- Purpose:* Executes a specified CI command synchronously with respect to user interaction and terminal time-outs.
- Syntax:* POLL interval|OFF command
- interval|OFF
if a number, it is the approximate number of minutes between executions of the poll command.
if OFF, the poll function is turned off.
- command is any CI command/program to be executed at the poll interval.

PR (Change Program Priority)*

Purpose: Change or display priority of a restored program.

Syntax: PR prog priority

prog Program name.

priority Range is between 1 and 32767.

PROT (Display/Change Protection)

Purpose: Display or change the protection status of a file or directory.

Syntax: PROT mask|luV [newProtection]

mask File mask that includes all fields of the file descriptor and a qualifier.

luV A CI volume number followed by a 'V'.

newProtection

Access allowed for owner, group, and other users (r = read, w = write). If omitted, the current protection status is displayed.

Syntax: owner[/group]/others

PU (Purge Files)

Purpose: Purge files.

Syntax: PU mask [OK]

mask File descriptor.

OK Suppress user prompt.

PWD (Display Working Directory)

Purpose: Display present working directory.

Syntax: PWD

* This command can be entered in response to a BREAKMODE or CM prompt.

QU (Timeslice Quantum)*

Purpose: Display or modify current system timeslice quantum and program priority at which timeslicing begins.

Syntax: QU [quantum [limit]]

quantum New system slice quantum; value is in range 0 to 32767 milliseconds. Default is 1500.

limit Priority level at which timeslicing begins. Default is 50.

RETURN (Return from Command File)

Purpose: Return to the previous level of command file nesting or to the interactive mode.

Syntax: RETURN[,return1[,return2[,return3[,return4
[,return5[,return_s]]]]]]

return1 Integer return status indicating success or failure of command file. A value of zero indicates success; a nonzero value indicates failure. If omitted, return1 is set to zero.

return2-5 Four integer values made available to return additional status information. Each omitted parameter is set to zero.

return_s A string of up to 80 characters. If omitted, return_s is set to null.

All the parameters for the RETURN command are position dependent; therefore, you must include commas to mark the positions of any omitted parameters.

The values returned are available in the predefined variables \$RETURN1 through \$RETURN5, and \$RETURN_S, defined at the end of this section.

CI always executes a RETURN command when the end of a command file is reached, whether or not you included the command at the end of the file.

* This command can be entered in response to a BREAKMODE or CM prompt.

RN (Rename File, Directory or Subdirectory)

Purpose: Rename a file, directory or a subdirectory.

Syntax: RN mask1 mask2

mask1 Current name.

mask2 New name.

RP (Restore Program File)

Purpose: Establish a permanent program ID segment.

Syntax: RP file [prog] [opts]

file File name. The first five characters of the file name are used as the program name, unless the optional parameter is specified.

prog Program name to be used.

opts A character string that contains “C”, “P”, “T”, or both “C” and “P” to select the following options:

C Create a clone name if the specified or assigned name is already assigned to an RP'd program.

The program is cloned if:

- There is an active program with that name that is not a system utility.
- There is a dormant, temporary program with that name that is not a system utility.

P Create a permanent program ID segment that will not be released when the program terminates.

T Create a temporary program ID segment that will be released when the program terminates.

RS (Restart Session Progenitor)*

Purpose: Aborts and reschedules the session progenitor

Syntax: RS

The RS command can be used to restart a session progenitor (usually CI) that is not executing properly (for example, when CI becomes hung on a downed device). This command is especially useful if you OF the progenitor, as the session will terminate if it is the only program in the session.

The RS command is available only from the CM and BREAKMODE prompts.

RU (Run Program)*

Purpose: Immediately schedule a program for execution and wait for its completion.

Syntax: [RU] prog|file [parm*5]

RU Required only if the program name can be interpreted as a CI command or a command file name.

prog|file A five-character program name or a file descriptor.

parm*5 Parameters to be passed to the program. The maximum runstring length, including the implied RU and delimiter, is 80 characters. This can be five parameters or one long character string.

* This command can be entered in response to a BREAKMODE or CM prompt.

SET (Display or Define Variables)

Purpose: Display all positional, user-defined and predefined variables, or define a user-defined or predefined variable.

Syntax: SET [variable=string]

variable A string of up to 16 letters, digits, and underscores, not starting with a digit.

string A string of up to 83 characters.

SL (Display/Modify Session LU Information)*

Purpose: Displays or modifies the Session Switch Table (SST) for either a specified session LU or all session LUs.

Syntax: SL [session_lu [system_lu]]

session_lu Specifies the session LU to display or modify; if omitted, the entire Session Switch Table (SST) is displayed.

system_lu Specifies the system LU as follows:

number = system LU to which the session LU will point.

– = removes the mapping for the given session LU.

SS (Suspend Program)*

Purpose: Suspend an active program.

Syntax: SS [prog]

prog Name of an active program.

ST (Display Program Status)*

Purpose: Display status of a program.

Syntax: ST [program|partition#|0]

program Display status of this program.

partition# Display program occupying this partition.

0 Display name and partition number of currently executing program.

SZ (Display or Modify Program Size)*

Purpose: Display or modify program size information.

Syntax: SZ prog [size [msegsize]]

prog Program name.

size Program size in pages (for a non-VMA program) or EMA size (for an EMA program), not including PTE. Range is $2 \leq \text{size} \leq 1022$ for EMA size.

msegsize New MSEG size for EMA programs. Range is $1 \leq \text{MSEG size} \leq 30$.

* This command can be entered in response to a BREAKMODE or CM prompt.

TI (Display Time)*

Purpose: Display the system real-time clock.

Syntax: TI

The current system time is displayed in the following format: year, day (Julian), hour (24-hour format), minutes, and seconds.

TM (Display or Set System Clock)

Purpose: Display or set the system clock.

Syntax: TM [month day year hr:min[:sec [pm]]]

month	Jan to Dec
day	1 to 31
year	1976 to 2144
hr	0 (default) to 23
min	0 (default) to 59
sec	0 (default) to 59
pm	AM (default) or PM

TO (Display or Set Device Timeout)*

Purpose: Display or set timeout limit for a device.

Syntax: TO eqt [interval]

eqt LU number of device.

interval Timeout value for device LU (in 10-ms intervals).
 $0 \leq \text{interval} \leq 65534$. If interval = 0, device does not time out.

* This command can be entered in response to a BREAKMODE or CM prompt.

TR (Transfer to Command File)

Purpose: Transfer control to a command file.

Syntax: [TR] file [parm*9]

TR Transfer command. Required only if the file parameter can be confused with a CI command or a program (type 6) file descriptor.

file File containing CI commands.

parm*9 One to nine parameters can be specified. They replace the positional variables \$1 through \$9 in the command file. Default to zero-length strings.

UL (Unlock Shareable EMA Partition)*

Purpose: Unlock a shareable EMA partition.

Syntax: UL label

label Identifies a shareable EMA partition label. “WH,SH” lists available labels.

UNPU (Unpurge Files)

Purpose: Recover purged files.

Syntax: UNPU mask

mask A file mask that specifies the files to unpurge. A file can only be unpurged if its space has not been reused. “DL,@.@.P” lists purged files.

* This command can be entered in response to a BREAKMODE or CM prompt.

UNSET (Delete User-Defined Variable)

Purpose: Delete a user-defined variable.

Syntax: UNSET variable

variable String of up to 16 letters, digits, and underscores, not starting with a digit. The variable must exist.

UP (Up a Device)*

Purpose: Notify the system that a specified device is available.

Syntax: UP eqt

eqt EQT number of the device.

UR (Release Reserved Partition)*

Purpose: Release a reserved partition.

Syntax: UR partition

partition Partition number of the partition to be released.

VS (Display/Change VMA Program Size)*

Purpose: Display or change size of a restored VMA program.

Syntax: VS prog [lastpg]

prog Program name, up to five characters.

lastpg Specify the last page of VMA; range is between 31 and 65535. Actual VMA size will be one page greater than the value entered.

* This command can be entered in response to a BREAKMODE or CM prompt.

WD (Display/Change Working Directory)

Purpose: Display or change the working directory.

Syntax: WD [directory [file | +s]]

directory Name of new working directory.

file Command stack file.

+s Post current command stack file.

WH (System Status Reporting)*

Purpose: Report system status information.

Syntax: WH [parm]

parm Default is user's session program:

AL Display status of all suspended and scheduled programs.

SM Same as AL except, state 3 programs without "parent-child" relationships are not listed.

PA Display status of all partitions being used.

PL|PR
Display all ID segments.

PL|PR [program]
Display information on only the program specified, up to 5 characters. May use dashes (–) as a wildcard character to match any program name character.

* This command can be entered in response to a BREAKMODE or CM prompt.

WHILE-DO-DONE (Control Structure)

Purpose: Allow repeated execution of a group of commands. WHILE-DO-DONE can be used only in a command file.

Syntax: WHILE list1
DO command-list2
DONE

command-list*n*

A list of commands either one command per line or multiple commands per line separated by semicolons. A command list can be null.

WHOSD (Report User of Directory or Volume)

Purpose: Report the session that is using a specified directory or directory on a specified volume as a working directory or as part of a UDSP.

Syntax: WHOSD [-t] [-m idmask] file|directory|lu

-t Trace ID segments back to a file name.

-m idmask

Check only those ID segments whose names match the mask.

file|directory|lu

Specifies the file, directory, or volume of which you want to report the user.

WS (Display or Modify VMA Working Set Size)*

Purpose: Display or modify VMA working set size of a restored program.

Syntax: WS prog [wssize]

prog Program name.

wssize Working set size in pages (not including PTE).
 $2 \leq \text{wssize} \leq 1022$. Default is 31 pages.

* This command can be entered in response to a BREAKMODE or CM prompt.

XQ (Run Program without Wait)

Purpose: Schedule a program for execution, then return to CI.

Syntax: XQ prog|file [parm*5]

prog|file Program name or file descriptor.

parm*5 Parameters to be passed to the program. The total runstring has a limit of 80 characters.

?/HELP(Help)

Purpose: Display information on a CI command.

Syntax: ?/HELP [command]

command CI command. If the command parameter is omitted, a list of commands is displayed.

Command Stack Editor

<i>Purpose:</i>	Allow previously entered command lines to be displayed, edited, and re-entered.
<i>Syntax:</i>	<code>/[:][./ n][.pattern]</code>
:	denotes auto-execute mode.
<code>/./</code>	number of slashes specifies line number at which to start frame.
<code>n</code>	line number at which to start frame.
<code>.pattern</code>	select lines containing this pattern; pattern syntax is: <code>[^ { - @ <char> \ <char> } ...</code> <code>^</code> anchors pattern to start of command line. <code>-</code> matches any single character. <code>@</code> matches zero or more characters. <code><char></code> matches specified character. <code>\<char></code> matches specified character, allowing <code>-</code> or <code>@</code> to be entered. <code>\</code> quoting character.

The command stack editing mode commands are:

<code>ctrl-A</code>	Go to start of line where cursor is positioned.
<code>ctrl-Z</code>	Go to end of line where cursor is positioned.
<code>ctrl-P</code>	Display previous frame of selected lines.
<code>ctrl-F</code>	Display following frame of selected lines.
<code>ctrl-K</code>	Mark current line for grouped execution in order of marking.
<code>ctrl-D</code>	Delete current line from stack.
<code>ctrl-Q</code>	Quit stack mode, start executing marked lines.
<code>ctrl-U</code>	Enter instead on <i>ctrl-Q</i> on terminals using XON/XOFF handshake protocol.

`ctrl-Q ctrl-Q`, `ctrl-U ctrl-U`

Abandon stack mode, forget marked lines.

\$1 – \$9 Positional Variables

Nine positional variables can be passed via the TR command to a command file. The parameters in the TR command are stored in variables 1 to 9. They are recalled by \$1 to \$9 in the command file.

Predefined Variables

When you begin a CI session, there are predefined variables. These variables are initialized to default values by CI. However, you can use the SET command to modify the values of all the variables except \$MY_NAME. Also, \$WD is different in that you can modify it, but CI updates it after each WD command. The SET and ECHO commands are used to display the values of predefined variables. You cannot delete these variables.

\$AUTO_LOGOFF

Allow for automatic logoff if session is inactive. CI initializes \$AUTO_LOGOFF to zero, which means that automatic logoff is not in effect. If you set \$AUTO_LOGOFF to a nonzero value, CI times out after that many terminal timeouts. If CI is the only active program, after four CI timeouts, an EX command is executed to terminate the session.

\$DATC

The datecode revision of the operating system; for example, 6000 for Revision 6.0. This variable is for user information and can be deleted by the UNSET command.

\$FRAME_SIZE

Size of command stack display. When you logon, the command stack display size is initialized to 20 lines. It can be set to any positive integer greater than zero.

\$HOME

\$HOME is set to the directory in which CI starts up. \$HOME cannot be deleted by the UNSET command.

\$LOG

A flag indicating if commands executed in a command file are logged to the terminal. CI initializes this variable to OFF, which means that commands are not displayed at the terminal. To display commands at the terminal, set the value to ON.

\$LOGON

The user and group name with which the user is associated during the session in the form USER.GROUP.

\$MY_NAME

The true or schedule name of CI. This variable can never be altered.

\$OLDPWD

\$OLDPWD is set to the previous working directory (\$WD) whenever a WD or CD command is executed.

\$OPSY

The ID number of your operating system.

\$PROMPT

The prompt that is displayed when CI is waiting for input. CI initializes this variable on the name of the program file containing CI.

\$RETURN1 – \$RETURN5

Five integer values (ASCII representation) returned from execution of the last command. CI updates the values as commands are executed. These variables can be set to values between -32768 and 32767 , inclusive.

\$RETURN_S

An 80-character string returned from execution of the last command. CI updates the value as commands are executed.

\$RU_FIRST

Flag indicating whether RU or TR is to be assumed if you only enter a file name in response to a CI prompt or as a line in a command file. CI initializes this variable to TRUE, which means CI first attempts to execute an RU command for the specified name. Set this value to FALSE if you want CI to assume that the file name entered is the name of a command file. You should set the variable to FALSE if you will be executing more command files than program files.

\$\$SAVE_STACK

Flag indicating if the command stack is saved when you exit CI or when the command stack file is changed with the WD command. CI initializes this variable to TRUE, which means the stack should be saved. Set the value to FALSE if you do not want the stack saved.

\$\$SESSION

Number of your current session. CI initializes this variable to your session number and updates the value after execution of every CI command.

\$WD

Name of the current working directory. CI updates this variable after execution of each WD command.

User-Defined Variables

A user-defined variable name is a string of up to 16 letters, digits, and underscores not starting with a digit. The SET command defines both the variable name and its contents. When you reference the variable after it has been defined, precede the name with a \$ (dollar sign) and follow it with a space or any other character that is not a letter, a digit, or an underscore.

FMGR Commands

Scheduling FMGR	3-1
AC	3-1
AN	3-2
CA	3-2
CL	3-2
CN	3-3
CO	3-4
CR	3-4
CS	3-6
CT	3-6
DC	3-7
DL	3-7
DP	3-7
DU	3-8
EX	3-8
HE	3-9
IF	3-9
IN	3-10
LI	3-10
LL	3-11
LO	3-11
MC	3-11
ME	3-11
OF	3-12
PA	3-12
PK	3-12
PU	3-13
RN	3-13
RP	3-13
RT	3-14
RU	3-14
SE	3-15
SL	3-15
SM	3-17
SP	3-17
ST	3-18
SV	3-19
SY	3-19

TE	3-19
TR	3-20
VL	3-20
WH.....	3-20
??	3-21
**	3-21
Command Stacking	3-21

Scheduling FMGR

Purpose: Manipulate files on FMGR directories. Perform operations that are not available from CI.

Syntax: RU,FMGR[,namr[,list[,severity code[,log]]]]

namr File name or LU containing command input.
list LU of list device (default = LU 1).
severity Display commands and error codes:

Code

- 0 Display all commands and error codes.
- 1 Display no commands, all errors.
- 2 Display no commands, no errors except those requiring response. Terminates job on serious error.
- 3 Same as 2 except job not terminated.
- 4 Display no commands, no errors, and do not abort job.

log LU of log device (default = input or LU 1).

AC,crn[,P|G[,size[,id[,#dir tracks]]]]

Capability: 10

Allocate a cartridge to the session user from the spare cartridge pool.

crn Cartridge Reference Number (CRN) to be assigned to the allocated cartridge.
P|G Private (P) or group (G) cartridge designation (default = P).
size Number of tracks needed on cartridge.
id ASCII identifier of cartridge (default = DC00XX; XX is system LU number of terminal).
#dir tracks Number of tracks used by file directory (default=1).

AN,message

Capability: 20

Print message on list device.

message Message to be sent to list device; maximum length of 72 characters following FMGR colon (:) prompt.

CA,global#[,val1[op1,val2[...op(n),val(n+1)]]]

Capability: 40

Calculate global parameter values.

global# Integer preceding G in G-type global, or “integer:P” for P-type globals.

val1-val(n+1) Values used in calculations; if omitted, global# is nulled.

op1-opn Operations performed on operands p1-p(n+1):

+ Add two operands.

– Subtract second operand from first.

/ Divide second operand by first.

* Multiply two operands.

O OR

X XOR (exclusive OR).

A AND

CL[AL]

Capability: 10

Display list of user accessible cartridges.

AL Display list of all cartridges in system.

CN[,namr[,function[,subfnctn]]]

Capability: 20

Issue control request to non-disk device.

namr	Type 0 file name or LU (default = LU 8).
function	Control code, mnemonic or octal (see EXEC 3 call). mnemonic RW Rewind (default for magnetic tape, terminal cartridge units, and mass storage devices). EO End-of-file. TO Top-of-form (default for line printer and terminal CRT). FF Forward space file. BF Backspace file. FR Forward space record. BR Backspace record. LE Leader (default for paper tape punch).
subfnctn	Carriage control. 0 Suppress spacing on next print operation only. + <i>n</i> Space <i>n</i> lines before next print operation. - <i>n</i> Page eject on lineprinter or space <i>n</i> lines on terminal.

CO [**,cartridge1**]
,namr],**cartridge2**[,**options**[,**name1**
,name2[,**msc**]]]

Copy all or selected files from an active cartridge to active cartridge 2.

Capability: 20

namr	File name, security code, and crn or mask. Minus signs (-) can be used as place holders.
cartridge1	Source cartridge, positive crn or negative LU.
cartridge2	Destination cartridge; positive crn or negative LU.
options	Copy options: C Clear destination cartridge. D Dump mode. E Eliminate extents. P Purge source files after copy. V Verify.
name1	Starting file name.
name2	Ending file name.
msc	Master security code.

CR,namr

Capability: 20

Create a disk file of specified type and size (no data is transferred).

namr	File type and size must be specified as greater than 0; record size must be specified for type 2 files.
------	---

CR,namr,lu ,REad ,WRITE ,Bspace ,Fspace ,BOth ,EOf ,LEader ,PAge ,cntrl ,BInary ,AScii ,cntrl

Capability: 20

Create a non-disk (type 0) file; data not transferred.

- namr File name, security code, and crn.
- lu LU of non-disk device (positive).
- REad Legal input/output (no default).
- WRite
- BOth
- Bspace Legal spacing (default = FS for READ devices, no
- Fspace space all others).
- BOth
- EOf Control subfunction (default = EO for mass storage
- LEader devices, LE for paper tape punch, PA for line
- PAge printer).
- cntrl
- BInary Type of data (default = AS).
- AScii
- cntrl

CS,lu,attribute[,outlu[,priority]]

Capability: 30

Modify or change spool options set by SL command.

lu	LU defined at setup.
attribute	One of the following:
RWind	Reset file to first record.
PURge	Change SAve flag to PURge.
SAve	Change PURge flag to SAve.
PASS	Remove HOld option.
END	Write EOF and terminate spool. Spool file placed in outspool queue (default).
BUffer	Change to buffering.
NBUffer	Change to no buffering.
NPass	Change LU and/or priority information, by specifying the two additional parameters, outlu and priority.
outlu	New LU (specify only if NPass is used).
priority	New priority (specify only if NPass is used).

CT,namr[,function[,subfnctn[,message]]]

Capability: 20

Issue control request to terminal.

namr	Type 0 file or terminal LU number (may be greater than 64).
function/ subfnctn	Octal code:
11B	Space down a specified number of lines. Subfunction: 0 skip 2 lines +n skip <i>n</i> lines -n skip <i>n</i> lines
20B	Enable terminal (default).
21B	Disable terminal.
22B	Set timeout. Subfunction: value in units of 10 msec.
message	Message to be written to the terminal.

DC,cartridge[,RR]

Capability: 10

Logically remove a cartridge from session user's environment by setting inactive bit in session control block. Non-session, deletes entry in system cartridge list.

- | | |
|-----------|---|
| cartridge | Positive Cartridge Reference Number (CRN) or negative LU. |
| RR | Session only; deletes cartridge entry in system cartridge list. |

DL[,cartridge[,security]]

or

DL,filedes[,security]

Capability: 10

List the file directory of one or all of the mounted cartridges.

- | | |
|-----------|--|
| cartridge | Cartridge Reference Number (CRN), positive for label or negative for LU. Zero or none specified lists all. |
| filedes | Mask specifying the file entries in the directory to be output. Minus signs (-) can be used as place holders for more flexibility. |
| security | Two-character FMP master security code. |

If the master security code is 0, default in command will not obtain long list showing security codes; a code (any code) must be supplied.

DP[,p1[,p2[,p3...[,p14]]]]

Capability: 20

Display parameter value or global names.

- | | |
|--------|---|
| p1-p14 | Parameter values or global names to be displayed. |
|--------|---|

DU,namr1,namr2 [,record format [,eof control [,file#[,#files]]]]

Capability: 20

Transfer data from an existing file or LU to another existing file or LU.
Does not create namr2.

namr1	Source of data.
namr2	Destination of data.
record format	Format of data or EOF control (default = namr1 format, or ASCII if non-disk device).
ASCII	ASCII records.
BReloc	Binary relocatable records with checksum.
BNary	Binary records without checksum.
BAbs	Binary absolute records with checksum.
MTape	Magnetic tape ASCII records.
MS	Magnetic tape SIO (System Input/Output) records are written on namr2. Standard records are expected on namr1.
MSBR	Magnetic tape SIO binary relocatable re- cords (same as MS+BR).
MSBA	Magnetic tape SIO binary absolute records (same as MS+BA).
IHhibit	Inhibits EOF on namr2 and leader punch- ing.
SAve	Save embedded EOFs in namr1.
file#	File or subfile on namr2 where transfer starts (default = 1).
#files	Number of files to be transferred from namr1 (default = 1).

EX

Capability: 1

Terminate FMGR (non-session).

EX, $\frac{SP}{RP}$ [,RG[,KI]]

Capability: 1

Initiate logoff process (session).

SP RP	Save release private cartridges.
RG	Release group cartridge.
KI	Abort any active session programs.

HE[,keyword[,lu]]

Capability: 1

Detailed error code explanation.

- keyword Identifiers related to error code (session default=last error posted). Non-session, keyword must be specified.
- lu Device for explanation output (default = user's terminal).

IF,p1,operator,p2[,skip]

Capability: 40

Compare two values (usually globals) and skip a specified number of commands. Command not allowed from interactive device, must be in procedure file or batch job.

- p1,p2 Values to be compared.
- operator ASCII operators as follows:
 - EQ p1 = p2
 - NE p1 ≠ p2
 - LT p1 < p2
 - GE p1 ≥ p2
 - LE p1 ≤ p2
- skip Number of commands to skip (positive or negative). Use -2 to skip back to previous command (default = 1).

IN,msc,crtrdge,lbl,id[,1st trk[#dir trks [,#sec/trk]]]

Capability: 60

Initialize a cartridge.

msc	Master security code; ignored in a session environment.
crtrdge	Cartridge Reference Number (CRN), positive for label or negative LU. (Must be $-LU$ if new.)
lbl	New cartridge reference label; $0 < lbl < 32768$, or two ASCII characters.
id	Cartridge information label; up to 6 ASCII characters, must follow FMGR namr restrictions.
1st trk	First track to be used on the cartridge. If LU 2, must be 8 greater than last system track (default = track 0).
#dir trks	Number of directory tracks (1 to 48), (default = 1).
#sec/trk	Number of 64-word sectors per track. If LU 2/3, parameter is ignored.

IN,master security code – new security code

Capability: 60

Change master security code. New code is separated from old code by two minus (-) signs.

LI,namr[,format[,ln1[,ln2]]]

Capability: 10

List contents of a file or LU on list device.

namr	File name or LU number.
format	Specify list format. S ASCII source (default for type 0, 3, 4 files). B Binary (default for all other type files). D Directory information only.
ln1	Starting line.
ln2	Ending line.

LL,namr

Capability: 20

Change current assignment of list device.

namr File name or LU number.

LO,lu

Capability: 50

Change LU number of log device.

lu LU number of new log device.

MC,lu[,P|G[,size[,id[,#dir trks[,label]]]]]]

Capability: 10

Make an unmounted cartridge available for use.

lu LU number of cartridge to be mounted, it must be in user's Session Switch Table.

P|G Private or group cartridge (session default=P) non-session meaningless, but its space must be provided.

size # of tracks needed on cartridge.

id ASCII identifier of cartridge (default DC00XX; XX is system LU number of terminal).

#dir trks # of tracks used by the file directory (default = 1).

label Cartridge Reference Number (CRN) to be assigned to the cartridge.

ME[,namr[,clear]]

Capability: 10

Display contents of user's message file.

namr File name or non-disk LU to receive messages (default = user's terminal).

clear 1 clear message file.
0 do not clear = default.

OF,program

Capability: 30

Terminate program in caller's current session.

program Name of program to be terminated.

OF,program

Capability: 60

Terminate any program in the system.

program Name of program to be terminated.

PA[,lu[,message]]

Capability: 40

Suspend execution of the current job or procedure file, and transfer control to a specified device, and optionally print a message.

lu LU to which control transfers (default = log device).
message 1-80 ASCII characters.

PK[,cartridge]

Capability: 20

Recover tracks and directory entries assigned to purged files and close gaps between files.

cartridge Cartridge Reference Number (CRN), positive for label or negative for LU (default = all user accessible cartridges).

PU,namr

Capability: 20

Remove a file and its extents from system.

namr File name, and, optionally, security code and CRN of existing file.

RN,namr1,namr2

Capability: 20

Change a file name and attributes.

namr1 Existing file name and optionally, security code and CRN.

namr2 New name, file type, and/or security code. No other subparameters may be changed.

RP,namr,program[,pname]

Capability: 30

Restore a program by assigning an ID segment to a type 6 file.

namr Name of a type 6 file that was saved with SP.

program Name of program whose ID segment is assigned to the restored program.

pname Name of the restored program.

RP,namr[,pname]

Capability: 30

Restores a program by assigning a new ID segment to the type 6 file.

namr Name of a type 6 file that was saved with SP.

pname Name of the restored program.

RP,,program

Capability: 30

Release a previously assigned ID segment.

program Name of program whose ID segment is to be released.

RT,program

Capability: 30

Release all disk tracks assigned to a dormant program.

program Name of program whose assigned tracks are to be released; must be dormant.

RU,program:IH[,parameters]

Capability: 30

Schedule a program for immediate execution and inhibit automatic renaming feature.

program Program name.
parameters Parameters or string to be passed to program.

RU[IH],program[,parameters]

Capability: 30

Schedule “program” for immediate execution. IH inhibits passing of command string.

program Name of program to be executed or namr of type 6 file containing program.
parameters 1-5 parameters to be passed to program.

SE[,p1[,p2[...[,p9]]]]

Capability: 40

Set or clear global parameters 1G-9G. If all parameters are omitted, globals are nulled. If any parameter is omitted, the corresponding global parameter is unchanged.

p1-p9 Values to be converted to global parameters 1G through 9G.

SL[,lu]

Capability: 10

Display linkage information for session logical unit number.

lu Session LU number (default = list information for all session LUs in user's Session Switch Table).

SL,lu[,namr[,attribute[,outlu[,priority[,prog]]]]]

Capability: 30/50

Set up spool file for an I/O device.

- lu Session LU to which a spool file is to be associated. The LU must not be LU 2 (system disk). LU 3 (auxiliary disk), any LU associated with a disk driver, a spool LU, or if in a job system LU 5 (standard spool input device).
- namr Name of existing file to be used as a spool file (default = system assigned spool pool file).
- attribute Define characteristics of spool access. Any 3 attribute codes can be combined, no delimiters necessary.
 Attribute codes:
 NO = Queue file for immediate outspool
 RE = Read only
 WR = Write only
 BO = Both read and write
 WN = Write now
 BU = Buffered
 PU = Purge
 SH = Write spool headers
 ST = Standard file format
 Default for attribute codes:

	namr not specified		namr specified	
outlu specified	WR	HO	WR	HO
	SH	SP	SH	SA
outlu not specified	BO	HO	RE	HO
	ST	SP	ST	SA

- SP = Spool pool file
- SA = Save (do not purge)
- HO = Hold until close

- outlu Session LU for outspooling.
- priority Outspool priority (default = session – 99, batch – priority of job).
- prog If specified, program “prog” will be scheduled with wait, by the spool system when spool LU is closed. Note the spool file will not be outspooled, “prog” must properly dispose of the file. Capability of 50 is required.

SL,session lu,system lu

Capability: 30/50

Define a session LU to be entered to or deleted from the user's Session Switch Table. Capability of 50 is required to add a system LU to the user's SST.

- session lu LU by which session users address a system LU.
- system lu May be specified as – (a dash) to delete LU mappings that have been created during user's session.

SM,user,namr,message

Capability: 10

Send message and/or file to another user's message file.

- user Logon ID of message recipient, (user.group).
- namr Name of file or non-disk LU containing data to be sent
- message String entered from sender's terminal.

SP,namr [,PR ,GR [,capability]]

Capability: 30

Place a disk-resident program and its ID segment in a type 6 file created by this command.

- namr File name of the type 6 file. First 5 characters of file name must be identical to disk program name. File subparameters default to:
 - security 0
 - cartridge First cartridge in user's cartridge list file type
 - Type 6
 - file size Size of program
 - record size 128
- PR Only users using the SP command can run or RP the type 6 file.
- GR Only users belonging to the same group as the user who issued the SP command can run or RP the type 6 file.
- capability Minimum capability level required to run or restore (RP) the type 6 file.

ST,namr1,namr2[,record format[,eof]][,file# [,#files]]]

Capability: 20

Transfer data from an existing file or LU to another file or LU.

namr1	Source of data.
namr2	Destination of data. If it is to be a disk file, it will be created.
record format	Format of data or EOF control (default = namr1 format or ASCII if non-disk device). ASCII ASCII records. BReloc Binary relocatable records with checksum. BNary Binary records without checksum. BAbs Binary absolute records with checksum. MTape Magnetic tape ASCII records. MS Magnetic tape SIO (System Input/Output) records are expected on namr1. Standard records are written on namr2. MSBR Magnetic tape SIO binary relocatable records (same as MS+BR).
eof	EOF control: IHibit Inhibit EOF on namr2 and leader punching. SAve Save embedded EOFs in namr1.
file#	File or subfile on namr1 where transfer starts (default = 1).
#files	Number of files to be transferred from namr1 (default = 1).

SV,severity[,global #[,IH]]

Capability: 20

Change the system log device severity code to a new number.

severity	New severity code.
0	Display all commands and errors (default).
1	Display no commands, all errors.
2	Display no commands, no errors except those requiring response. A serious error terminates job.
3	Display same as 2, except job not terminated.
4	Display no commands, no errors, job not terminated.
global #	Optional G global number (1-9) into which current severity code is to be placed.
IH	Optional parameter to inhibit echo of command entry.

SYcommand

Capability: 1

Execute RTE system command from FMGR.

command The system command mnemonic code. A delimiter between SY and command is not permitted.

TE,message

Capability: 10

Send message to the system console.

message Message to be sent to system console.

TR [**namr**, **-integer** [,parameters]]

Capability: 1

Transfer control to a file or LU, passing parameters as globals. A comma (,) or colon (:) may replace the TR, as the transfer command code.

- namr** File or LU to which control is transferred.
- integer** A negative integer that denotes a transfer back the specified number of files.
- parameters** The parameters to be set into the globals (1G-9G). Skipped parameters are not changed.

VL,**cartridge**

Capability: 60

Assign system scratch and VMA backing store cartridge.

- cartridge** Positive Cartridge Reference Number (CRN) or negative LU; default (command not entered or cartridge = 0) is /SCRATCH/ if it exists or first available cartridge in user's cartridge list.

WH[,lu[,option[,program]]]

or

WH[,option[,program]]

Capability: 10

Schedule WHZAT program.

- lu** The session LU for display.
- option** default User's session programs:
 - AL** Display status of all the suspended and scheduled programs.
 - SM** Same as AL except state-3 programs without "parent-child" relationships are not listed.
 - PA** Display status of all partitions being used.
 - PL or PR** Display status of all programs in the system.
- program** Used only with PL/PR option. Display only information on program specified.

??[error#]

Capability: 10

Request FMGR error code explanation.

error# FMGR error code (default = last error issued).

**Comment Line

Capability: 10

Includes lines of comments in a FMGR command entry list.

comment line
Any string of alphanumeric characters.

Command Stacking

- Ln* *n* is the number of lines to list (default is to list the entire command stack).
- P* Display or edit the pending line in the command stack. Edit options are CNTL/R, CNTL/I, CNTL/S, CNTL/T and CNTL/C. See the chapter on the Interactive Editor.
- n* Position pending line to the *n*th line in the command stack.
- ^n* or *Rn* Position *n* lines preceding pending line.
- /n* Position *n* lines past pending line.
- n* Delete *n* lines from command stack from the pending line.

Once a line has been displayed as the pending line, it may be executed by entering a carriage return.

System and Breakmode Commands

AB	4-1
AG	4-1
AS	4-1
BL	4-2
BR	4-2
CU	4-2
DN	4-3
EN	4-3
EQ	4-4
FL	4-4
GO	4-5
HE	4-5
IT	4-5
LU	4-6
OF	4-6
ON	4-7
OP	4-7
PR	4-7
QU	4-8
RS	4-8
RT	4-8
RU	4-9
SL	4-9
SS	4-9
ST	4-10
SZ	4-11
TE	4-12
TI	4-12
TM	4-12
TO	4-12
UL	4-13
UP	4-13
UR	4-13
VS	4-14
WH	4-15
WS	4-15

AB,optn

Capability: —

Abort currently executing batch job. Under session, the command is valid only when entered from the system console.

optn	0 Disk tracks not released.
	1 Release all disc tracks.

AG,numb*

Capability: 60

Modify partition priority aging rate.

numb	Partition priority aging rate; where numb is number of 10-ms intervals used as the aging rate.
------	--

AG,OF*

Capability: 60

Turn off partition priority aging.

AS,program,partition#*

Capability: 50

Assign a program to execute always in the same partition.

program	Name of program to be assigned to specified partition.
partition#	Partition number to which program is to be assigned. If 0, program is unassigned.

* This command also available in CI.

BL*

Capability: 10

Examine current buffer limits.

BL[,lower[,upper]]*

Capability: 60

Modify current buffer limits.

- | | |
|-------|---|
| lower | Lower buffer limit specified in number of words (default = 1). |
| upper | Upper buffer limit specified in number of words (default = existing limit). |

BR[,program]

Capability: 10/60

Set break flag for any program in user's session. User program tests for a set break flag with subfunction I=IFBRK (DUMMY).

Capability level of 10 or greater allows user to set break flag in session program. Capability level of 60 allows user to set break flag of any program in system.

- | | |
|---------|--|
| program | Name of program whose break flag is to be set; default is current session program. |
|---------|--|

CU,^{ON}_{OFF} *

Capability: 60

Turn the CPU utilization (S-Register) display on or off.

* This command also available in CI.

DN,,lu

Capability: 60

Down I/O device.

lu System logical unit.

DN,eqt*

Capability: 60

Down I/O controller.

eqt Equipment Table (EQT) entry number.

EN,msc[,option]

Capability: —

Enable system console as a session terminal. Command only valid when entered from the system console.

msc	Two-character FMP master security code.
option	0 Master security code not required in OP commands (default).
	1 Master security code is required in O commands.

* This command also available in CI.

EQ,eqt*

Capability: 10

Print description and status of an I/O controller.

eqt Equipment Table (EQT) entry number of an I/O controller.

Status information is printed as:

select code DV.nn D B Unnn status

select code

Is the I/O select code number.

DV.nn Is the driver routine.

D Is D if DMA is required; 0 if not.

B Is B if automatic output buffering; 0 if not.

Unnn Is the last subchannel addressed.

status Is the logical status.

0 = Available.

1 = I/O controller down.

2 = I/O controller busy.

3 = Waiting for DMA assignment.

EQ,eqt,UN BU *

Capability: 60

Change the automatic buffering designation for a particular I/O device.

eqt Equipment Table (EQT) entry number of an I/O controller.

UNbuffer Turn off buffering.

BUffer Turn on buffering.

FL

Capability: 10

Eliminate buffered output to a session terminal. Only valid in break mode, and not valid from system console.

* This command also available in CI.

GO[IH][,program][,p1[...[,p5]]]

Capability: 30/60

Reschedule program previously suspended by the SS command or a suspend EXEC call. GOIH inhibits passing of command string. Capability level of 30 allows user to reschedule programs in their sessions. Capability level of 60 allows user to reschedule any program in the system.

- program Name of suspended program to be rescheduled (default = current session program).
- p1-p5 Parameters to be passed to program; only passed if program suspended itself with EXEC call.

HE[,keyword[,lu]]*

Capability: 1

Detailed error explanation.

- keyword An eight-character error code (default = last error logged).
- lu Device to which explanation is output (default = user's terminal).

IT,program,res,mpt[,hr[,min[,sec[,ms]]]]*

Capability: 50

Set automatic execution time value for a program. ON command must follow to schedule the program. Not specifying optional parameters removes program from the time list (program must be dormant).

- res Resolution code:
 - 1 tens of milliseconds
 - 2 seconds
 - 3 minutes
 - 4 hours
- mpt Multiplier (0-4095) used with res.
- hr,min, Initial start time.
- sec,ms seconds, milliseconds

* This command also available in CI.

LU,lu*

Capability: 60

Print EQT entry number, device subchannel number, associated with a system LU, and whether the device is up or down. See SL command for similar function.

lu System LU.

LU,lu,eqt[,subchan#]*

Capability: 60

Reassign new EQT entry number to system LU. If EQT number has subchannels, use the subchannel parameter.

lu System LU.

eqt Equipment Table (EQT) entry number to assign LU.
If 0, LU becomes the system bit bucket.

subchan# Subchannel number to assign LU.

OF[,program[,numb[,NP]]]*

Capability: 30/60

Terminate a program. Capability level of 30 allows user to terminate any program in the current session. Capability level of 60 allows user to terminate any program in the system.

program Name of program to be terminated. If not specified, an "OF,program,1" is executed for current session program.

numb 0 Remove from time list; disk tracks not released (default).

1 Terminate immediately; release disk tracks.

ID or 8

Terminate immediately and permanently from system (must be issued to segments as well as the main).

NP Abort message is not printed.

* This command also available in CI.

ON[IH],program[,NOW][,p1 [...[,p5]]]*

Capability: 50

Schedule a program for execution. Program's entry in time list is affected. ONIH inhibits passing of command string.

- program Name of program to be scheduled.
- NOW Schedule program immediately.
- p1-p5 1-5 parameters passed to program when it is scheduled.

OP[,msc[,command]]

Capability: —

Enter a system level command from a low capability session. Command only valid when entered from the system console.

- msc Two-character FMP master security code. If specified in the EN command the security code is required.
- command The system command to be executed.

PR,program,priority*

Capability: 50

Change the priority of a program.

- program Name of program whose priority is to be changed.
- priority New priority; 1 is the highest priority and 32767 is the lowest.

* This command also available in CI.

QU[,quantum[,limit]]*

Capability: 10/60

Examine system timeslice quantum and fence. Capability level of 60 is needed to change the timeslice parameters.

quantum System timeslice quantum, value 0-32767 milliseconds (default = 1500).

limit Priority level fence to begin timeslicing (default = 50).

RS

Capability: 10

Abort and reschedule a session's copy of FMGR or CI.

RT,program

Capability: 30

Release all disk tracks assigned to a program.

program Name of program whose assigned local disk tracks are to be released.

* This command also available in CI.

RU[*IH*],program[*p1* [...*p5*]]*

Capability: 30

Schedule a program for immediate execution. Program's entry in time list is not affected. RUIH inhibits saving of command string. The breakmode RU actually runs the specified program, not a renamed copy of the program.

program Name of program to be run.
p1-p5 Parameters passed to program when it is scheduled.

SL[,*lu*]*

Capability: 10

Display session LU information.

lu Session LU for which linkage information is desired.
(Default is to list information for all session LUs in the user's Session Switch Table.)

SS[,program]*

Capability: 30/60

Suspend non-dormant session program. Capability level of 30 allows users to suspend any program that was scheduled within their session. Capability level of 60 allows users to suspend any program in the system. If program name is not specified, the current session program is suspended.

program Name of program to be suspended.

* This command also available in CI.

ST[,program]*

Capability: 10

Determine status of program.

program Program name.

The status is printed as follows:

```
pr S res mpt hr min sec ms T
```

pr Decimal priority number.

S Current status program:

- 0 Dormant.
- 1 Scheduled.
- 2 I/O suspended.
- 3 General wait.
- 4 Unavailable memory suspend.
- 5 Disk allocation suspend.
- 6 SS or EXEC 7 suspend.
- 9 Background segment.

res/mpt/
hr/min/
sec/ms 0 or time program is next scheduled to run.

T Program currently in time list.

ST,numb*

Capability: 10

Determine name or partition number of program currently executing.

numb 0 Display name and partition number of program currently executing in memory. 0 displayed if none executing.

Partition #
Display name of program currently residing in that partition. 0 if none.

* This command also available in CI.

SZ,prog*

Capability: 30

Display the program's size information.

prog Program name

The program size information is formatted as follows:

```
AAAAA BB CCCC DDDD EE FFFFF
```

AAAAA	The logical address of the last word plus 1 of the program.
BB	Required program size in pages.
CCCC	Minimum required partition size in pages. Program code + EMA. For MLS programs CCCC is the path length plus the desired dynamic buffer space.
DDDD	Program's EMA or working set size (VMA programs only).
EE	MSEG size (EMA/VMA programs only).
FFFFF	Last VMA page used by the program (VMA programs only).

SZ,program,size[,mseg size]*

Capability: 30

Change size of program specified.

program Program name.
size Non-EMA program: required program size. EMA program: required EMA size.
mseg size New MSEG size (non-shareable EMA program only).

* This command also available in CI.

TE,message

Capability: 10

Send message to system console.

TI*

Capability: 10

Print current year, Julian day and time.

TM,year,day[,hr[,min[,sec]]]**

Capability: 60

Set real-time clock.

year	Four digits (for example, 1957).
day	Three-digit Julian date (for example, 063 = March 4).
hr	Two-digit hour of the day (default is 0).
min	Two-digit minute of the hour (default is 0).
sec	Two-digit second of the minute (default is 0).

TO,eqt[,numb]*

Capability: 10/60

Capability level of 10 allows the user to examine the current timeout value of an I/O controller. Capability level of 60 allows the user to change the timeout value.

eqt	Equipment Table (EQT) entry number of the I/O controller.
numb	Number of 10-ms intervals to be used as the timeout value in the range 0 to 65534. Numb cannot be less than 500 ms for terminals driven by drivers DVR00 or DVR05.

* This command also available in CI.

** This command is different from the CI TM command.

UL,label*

Capability: 60

Unlock a shareable EMA partition.

label Label of the shareable EMA partition.

UP[,eqt]*

Capability: 10

Make I/O controller (and all associated LUs) available.

eqt Equipment Table (EQT) entry number of the I/O controller to be re-enabled. If not specified, re-enable the device (EQT number) that FMGxx or one of its child programs is waiting for.

UR,partition#*

Capability: 50

Release a reserved partition.

partition The number of the partition to be released (1 to 64, depending upon how many memory partitions were defined in system generation).

* This command also available in CI.

VS,program*

Capability: 30

Display the size for virtual memory data of a specified VMA program.

program VMA program name.

The virtual memory size information is formatted as follows:

AAAAA BB CCCC DDDD EE FFFFF

AAAAA The logical address of the last word plus 1 of the program.

BB Required program size in pages.

CCCC Minimum required partition size in pages (program code and working set).

DDDD The program's working set size in pages.

EE The program's MSEG size.

FFFFF The program's virtual memory size.

VS,program,lastpg*

Capability: 30

Modify the size for virtual memory data of a specified VMA program.

program VMA program name.

lastpg Requested last page of virtual memory size (default = 8191 pages).

* This command also available in CI.

WH[,lu[,option[,program]]]*

or

WH[,option[,program]]*

Capability: 10

Schedule WHZAT program.

lu	The session LU for display (default = user's terminal).
option	Default user's session programs.
AL	Display status of all suspended and scheduled programs.
SM	Same as AL except that state-3 programs without "parent-child" relationships are not listed.
PA	Display status of all partitions being used.
PL or PR	Display status of all programs in system; listing all the ID segments.
program	Used only with PL/PR option. Display only information on program specified.

WS,program*

Capability: 30

Display the working set size of a specified VMA program (pages).

program VMA program name.

See the VS command for an explanation of the output.

WS,program,wrsz*

Capability: 30

Modify working set size of virtual memory program.

program VMA program name.

wrsz New working set size of program (default = 31 pages).

* This command also available in CI.

EDIT/1000 Commands

EDIT Runstring	5-1
Line Specification	5-1
Edit Session Options (SE Command)	5-2
Screen Edit Commands	5-3
Screen Mode Control Commands	5-3
Ctrl-D Mode (Line Graphic Editing)	5-3
Display Commands	5-5
Line Edits	5-7
Line Mode Character Edits	5-7
Search Commands	5-8
Search Options	5-8
Exchange Commands	5-9
Exchange Options	5-9
File Input/Output Commands	5-10
Termination Commands	5-10
Control Commands	5-11

RU,EDIT[,filedes[,commands]]

- filedes File descriptor of the file to be edited.
- commands Edit commands to be executed upon entering EDIT. Multiple commands are separated by a command separator “|” (default).

EDIT prompt character “/” (default).

Use a space or a comma to separate elements in the runstring or the command.

Line Specification

- . Current pending line.
- \$ or > Last line in the file.
- n Line number.
- [line spec1][line spec2]
 Line range specifications (absolute line numbers or an offset from the current line).
- * First line of line spec.
- ^n or -n Backward *n* lines.
- +n Forward *n* lines.
- :x Line mark A through line mark Z.
- 'pattern' Search forward to line containing specified pattern.
- 'pattern' Search backward to line containing specified pattern.

Use a space or comma between commands or parts of commands that might otherwise be ambiguous.

Edit Session Options (SE Command)

`SE,option[,value]` Sets various EDIT session options and defaults.

EDIT session options:

AC	anchor character (default= ^)
AS	verify dangerous commands (default=on)
BE	bell with prompt (default=off)
CD	set line editing graphics submode within screen mode (default=off)
CF	case folding (default=on)
CS	command separator (default=)
DF	display functions in screen mode (default=on)
EC	escape character (default=\\)
FL	fill columns: first, last, end (1, 70, 256)
IC	indefinite character (default=@)
IN	save indentation on fill (default=off)
LE	maximum characters on line (default=256)
PC	EDIT prompt character (default=/)
QU	quiet option (default=off)
RE	regular expressions (default=off)
RT	return to pending line after multiple search (default=on)
SD	screen size (default=10,10,2)
SL	screen length (default=30 or 100)
SW	screen width (sensed when needed; <code>SWn</code> sets screen width to <i>n</i>)
TC	tab character (default=TAB key or ctrl-I)
TS	time stamp update (default=on)
VW	vertical window (default=10,10)
WC	window columns (default=1,256)

Screen Edit Commands

[line spec1][[line spec2]S Start screen edit.

Screen Mode Control Commands

Control commands entered twice leave the workfile unchanged.

- ctrl-Q Save screen edits and quit screen mode.
- ctrl-U Same as ctrl-Q; use for terminals with Xon/Off handshake protocol.
- ctrl-P Go to previous screen.
- ctrl-R Same as ctrl-P; use for X.25 pad terminals.
- ctrl-F Forward one screen.
- ctrl-S Start next screen.
- ctrl-T Same as ctrl-S; use for terminals with Xon/Xoff handshake protocol.
- ctrl-X Start next screen with larger (maximum) screen size.
- ctrl-C Execute command in line mode and return to screen mode.
- ctrl-D Enter ctrl-D mode (see below).
- esc-4-ctrl-B Break line at cursor position.
- ctrl-J Join following line to this one.
- ctrl-O Duplicate line on the screen.
- ctrl-A Move to left end of current line.
- ctrl-Z Move to right end of current line.
- ctrl-K Mark line; enter alphabetic character at colon.

Ctrl-D Mode (Line Graphic Editing)

- A [*option*] Arrow. Draw a path through defined points and put an arrow head at end of path.
- B [*option*] Box. Draw a closed polygon through defined points.
- C Copy.
- E Erase path (same as P E command).
- L [*option*] Lines. Take point list as pairs of points and draw lines between the pairs.

- M Move. Same as copy, but erase original area.
- P [*option*] Path. Start at the first point in the list and draw a line through each consecutive point until end of the list is reached.
- Q Quit ctrl-D mode.
- R Re-mark. Restore marks of previous ctrl-D mode command and re-enter ctrl-D mode. If the previous command was a copy or move, the locator points are not restored.
- U Undo. Undo the last ctrl-D mode command.

where *option* can be one of the following:

- N No read. Can save time if no screen mode changes have been made.
- B Bold line style.
- E Erase.
- P Paired line style.

Display Commands

?[command]

or

H [command]

or

HE [command]

Display information about the specified command (default=summary of commands).

?[option]

or

H [option]

Display the requested information.

Options:

- AB listing of EDIT abort messages.
- DA revision code of the EDIT program being run.
- EX explanation of EDIT abbreviations.
- LS line specification explanation.
- PA pattern explanation.
- PL pending line edit information.
- RE regular expression explanation.
- RM recover mode explanation.
- RO description of EDIT runstring options.

When “[more . . .” is displayed at bottom of a help screen, pressing RETURN displays the rest of the current help information without pause. Type “a” to abort help listing. Pressing any other key causes the next help screen to display.

HL[P] Display ruler header line. P makes ruler same length as pending line.

?? Display current version of EDIT program and current file.

[line spec1][line spec2]**L**[max] [+][list file][/]
Display a maximum number of lines (default = 20).

The lines can be listed or appended (+) to a list file.
Enter '/' to suppress OK? prompt.

L Display 20 lines plus next pending lines.

[line spec1][line spec2]**LN**[max] [+][list file][/]
Display 20 lines with line numbers plus next pending line.

[line spec1][line spec2]**LU**[max] [+][list file][/]
Display lines without line numbers (turn off LN).

[line spec1]**LE**
Display the number of characters in the current pending line or specified line.

LI Display the number of lines in the file.

n Display line *n*, making it the pending line.

+*n* Forward *n* lines and display new pending line.

-*n* Go back *n* lines and display new pending line.

N Display the line number of the pending line.

SH[option] Display the specified EDIT option (if *option* is not specified, summary of all EDIT options is displayed). *option* may also be:

- FM show whether changes have been made.
- MA display current text marks.
- SW display current screen width.
- UN display the Undo List maintained by EDIT.

SH UN [*n1*, *n2*]
Display the commands that would be used by the UN command to undo the last change.

[line spec1]**SZ**
Display number of 16-bit words in the destination file.

[line spec1][line spec2]**W**
Display window of lines and return to pending line.

WN | **WU** Display window of 20 lines with line numbers (WN) or without line numbers (WU) and return to pending line.

/ Display command stack.

Line Edits

- [line spec1]**I**<line char edits>
Insert text before specified line (default is pending line).
{<space>} <text>
Add text after pending line.
- [line spec1][line spec2]**K**[max][+][list files][/]
Delete the specified lines (default is pending line). The deleted lines can be saved (or appended if +) to a list file. Enter '/' to suppress OK? prompt.
- [line spec1]**C**<line char edits>
Edit pending line then advance pending line.
- [line spec1]**O**<line char edits>
Duplicate the specified line and edit copied line.
Display edited line (default is pending line).
- [line spec1]**P**<line char edits>
Edit the specified line and display (default is pending line).
- [line spec1]**Q**
Edit pending line with terminal edit keys.
- [line spec1]**R**<text>
Replace the specified line with text (default is pending line).
- [line spec1]**J**
Join the specified line to the following line (default is pending line).

Line Mode Character Edits

- | | |
|--------|--|
| ctrl-B | Break line at cursor position. |
| ctrl-C | Delete characters. |
| ctrl-R | Replace characters. |
| ctrl-S | Insert characters. |
| ctrl-P | Insert characters (frees ctrl-S to be used with Xoff). |
| ctrl-T | Truncate line at cursor position. |
| ctrl-X | Extend line, adding characters to end of line. |
| \ | Escape character. |
| ctrl-\ | Non-printing escape character. |
| / | Enter current prompt (default = /) with any control character edit to preserve character in that position. |
| <tab> | Skip to next tab stop, leaving skipped characters unmodified. |
| <text> | Replacement text to be entered on pending line. |

Search Commands

[line spec1][line spec2]**B**/pattern/ [A,N,Q,V]

Find a line with pattern (default = last pattern specified) within the specified range (default = line 1 to EOF).

[line spec1][line spec2]**F**/pattern/[A,N,Q,V]

Find a line with pattern (default = last pattern specified) within the specified line range (default = pending line to EOF).

[line spec1][line spec2]**D**/pattern/ [A,N,Q,V]

Delete lines from the specified line (default = current line) to the line containing the specified pattern (default = last pattern specified).

' pattern Search forward for a line containing pattern.

` pattern Search backward for a line containing pattern.

Search Options

- A Multiple (All) search
- N No-window parameter
- Q Suppress display (Quiet)
- V Reverse match

Exchange Commands

[line spec1][line spec2]**G**/match pattern/substitute/[N,R,S]

Exchange 'match pattern' with substitute over the specified range without listing (default = pending line).

[line spec1]**Y**/match pattern/substitute/[N,Q,R,S]

Exchange 'match pattern' with substitute on the specified line and display the next occurrence (default = pending line).

[line spec1][line spec2]**U**/match field/substitute/[Q]

Unconditional exchange over specified range of number of characters in 'match field' with substitute.

[line spec1][line spec2]**X**/match pattern/substitute/[N,Q,R,S][/]

Exchange 'match pattern' with substitute over the specified range (default = pending line). '/' suppresses OK? prompt.

Exchange Options

- N No-window parameter
- Q Suppress display (Quiet)
- R Remove zero-length records
- S Single exchange parameter (one exchange per line)

File Input/Output Commands

FCL	Close the list file (opened with the K or L command).
FCS	Close the source file.
FI filedes [/]	Replace the work area with another file. '/' suppresses OK? prompt.
[line spec1] M filedes [start line]	[#lines] or [:stop line]
	Merge in file specified after pending line.
WC filedes	Create file without exiting EDIT session.
WR [filedes]	Replace (write over) file without exiting EDIT session. Default is current file.

Termination Commands

A [/]	Abort EDIT leaving source file unchanged. '/' suppresses OK? prompt.
AS [/]	Abort EDIT and save work file. '/' suppresses OK? prompt.
EC filedes	Create a file and end EDIT session.
ER	Replace (write over) source file and end EDIT session.
ER filedes	Replace (write over) specified file and end EDIT session.

Control Commands

<space>text Append a line of text following current pending line.

#xxx[*n1*,*n2*] Add sequence numbers with 3-character identifier field *xxx* in columns 73 through 80. *n1* is the starting number; *n2* is the increment number.

***<text>** Enter comment line in command file.

[*line spec1*][*line spec2*]**_**[*n*][**Q**]
Repeat command *n* times. ‘|’ is command separator and must precede the repeat command.

[*line spec1*][*line spec2*]**BC** startcol[:stopcol] [*destcol*] [**Q**]
Block copy.

[*line spec1*][*line spec2*]**BK**
Kill trailing blanks in specified line range.

[*line spec1*][*line spec2*]**BM** startcol[:stopcol] [*destcol*] [**Q**]
Block move.

[*line spec1*][*line spec2*]**CO** [**Q**]
Copy text.

[.] [*****] **FL** [**Q**] [**R**] [**I**]
Fill text. ‘R’ removes indentation; ‘I’ leaves indentation.

[*line spec1*]**Kx** Mark a line with character *x*.

[*line spec1*][*line spec2*]**MO** [**Q**]
Move text.

RU,program Run program and return to EDIT at pending line upon termination.

SC Copy screen memory; insert after pending line.

T[*n1*,...,*n30*] Set tab columns up to 30 settings.

TA Set tab columns to 7 and 21 (for Assembly).

TF Set tab columns to 7, then every 4 (for FORTRAN).

TL Set terminal tab stops to line mode tabs.

TM Set tab columns to 10,26,40,44,48 (for Macro).

TP Set tab columns to every 3 (for Pascal).

TS Set terminal tab stops to screen mode tabs.

TU Set tab columns to 8 (UNIX default).

[*line spec1*]**TI***n*
Add time and date to line starting at column *n*.

[*line spec1*][*line spec2*]**TK**
Expand tab characters into spaces.

- TR**,namr,[Q][/] Transfer to a command file or an input device. Specify Q to suppress the listing; specify / to suppress OK? prompt.
- UN** Undo the command executed immediately before this command and revert file to the prior state.
- [line spec1]**UY**[n1][n2] Return text removed via the Undo (UN) command to the text file.
- /[n] *or* Command stack.
//[...]

LINK Commands

LINK Runstring	6-1
AB (Abort)	6-1
AS (Assign Partition)	6-1
BG (Background)	6-1
CA (Capability)	6-2
CR (Specify Scratch File or Cartridge)	6-2
DB (DBUGR)	6-2
DC (Do Not clone)	6-2
DE (DEBUG)	6-2
DI (Display)	6-3
DM (Debug Monitor)	6-3
DP (Do Not Purge)	6-3
EB (Extended Background)	6-3
EC (Echo)	6-3
EM (Extended Memory Access)	6-4
EN (End)	6-4
FO (Force Load)	6-4
IF (Conditional)	6-4
LB (Large Background)	6-4
LI (Library)	6-5
LK (Relink)	6-5
LL (List Option)	6-5
LO (List Program Attributes)	6-5
MA (Display Load Map)	6-5
MS (Multiple Search)	6-6
NA (Name)	6-6
OR (Order EMA Area)	6-6
OS (Operator Suspend)	6-6
OU (Output)	6-6
PA (Page Align EMA Area)	6-7
PR (Priority)	6-7
PS (Page Align Overlays)	6-7
RC (Reverse Common)	6-7
RE (Relocate)	6-7
RM (Relocate Module)	6-8
RO (Reorder)	6-8
RT (Real-time)	6-8
SC (System Common)	6-8

SE (Search)	6-8
SH (Shareable EMA)	6-9
SN (Snapshot)	6-9
SS (Subsystem Global Area)	6-9
SZ (Size)	6-9
TR (Transfer)	6-9
VM (Virtual Memory Size)	6-10
WS (Working Set Size of VMA)	6-10
* (Comment)	6-10
? (Help)	6-10

LINK Runstring

parms Any number of parameters up to the 254-character runstring limit. Commands and/or file names can be specified in any order. A file name causes the file to be used in the linking process; how it is used depends on name conventions. A file name must include a file extension or start with a corresponding prefix as follows:

<u>Extension</u>	<u>Prefix</u>
.LOD	#
.REL	%
.MAP	"
.LIB	\$
.RUN	any alphabetic character
.SNP	\$ ^
.DBG	@

A command is prefixed with plus (+). Command parameters are delimited with colon (:). If no parameters are specified in runstring, LINK runs interactively, ending with the EN command. Interactive commands do not require the plus sign, and command parameters can be separated by either a space or a comma.

AB (Abort)

Purpose: Abort LINK immediately.

Syntax: AB

AS (Assign Partition)

Purpose: Assign a partition where the program will reside.

Syntax: AS partNum

partNum Decimal number between 0 (default) and 1023. A partNum of 0 cancels number previously specified.

BG (Background)

Purpose: Specify background (type 3) program.

Syntax: BG (+BG in runstring).

CA (Capability)

Purpose: Set capability needed to run program.

Syntax: CA,PR|GR|OF[,capability]

PR Private; only user can run the program.

GR Group; any user in group can run the program.

OF Off; only used during relinking to cancel the capability level that was set previously.

capability Capability level needed to run the program. If you omit this parameter, LINK allows any user to run the program.

CR (Specify Scratch File or Cartridge)

Purpose: Specify file or cartridge (CRN) to be used for LINK scratch files. CRN is used only when running LINK from FMGR. Default is a scratch file created by LINK on the working directory.

Syntax: +CR:file|crn (in runstring only).

DB (DEBUGR)

Purpose: Append DEBUGR to program.

Syntax: DB

DC (Do Not clone)

Purpose: Inhibit cloning of a program.

Syntax: DC (+DC in runstring).

DE (DEBUG)

Purpose: Create .DBG file for Symbolic Debug/1000.

Syntax: DE (+DE in runstring).

DI (Display)

Purpose: Display undefined externals.

Syntax: DI

DM (Debug Monitor)

Purpose: Turn on debug monitor mode.

Syntax: DM (+DM in runstring).
DM[,OF] (relink syntax)

OF Turn off debug monitor mode; valid only during relinking.

DP (Do Not Purge)

Purpose: Inhibit purging of existing program files when running LINK interactively. EN or NA overrides DP.

Syntax: +DP (in runstring only).

EB (Extended Background)

Purpose: Specify that the program to be linked is an extended background program.

Syntax: EB (+EB in runstring).

EC (Echo)

Purpose: Echo loader command file commands.

Syntax: EC (+EC in runstring).

EM (Extended Memory Access)

Purpose: Set EMA size.

Syntax: EM pages

pages Number of pages of EMA space, in range 2 to 1022.

EN (End)

Purpose: End command input.

Syntax: EN [file]

file File descriptor of program output file. Can be defaulted to file descriptor specified by OU command, source PROGRAM (NAM in Macro) statement.

FO (Force Load)

Purpose: Force-load program or overlay, regardless of undefined externals.

Syntax: FO

IF (Conditional)

Purpose: Selective execution for specified system.

Syntax: IF A|6 linkcommand.

A Execute when linking on or for RTE-A.

6 Execute when linking on or for RTE-6/VM.

linkcommand
A valid LINK command.

LB (Large Background)

Purpose: Specify large background program.

Syntax: LB (+LB in runstring).

LI (Library)

Purpose: Define library file that the loader searches immediately before searching snapshot file and system libraries. Up to 10 library files can be defined by repeating this command.

Syntax: LI file
file File descriptor of file to be searched.

LK (Relink)

Purpose: Change attributes of previously linked program. (LO command lists current program attributes.)

Syntax: LK file
file File descriptor of program file to be relinked.

LL (List Option)

Purpose: Specify destination for list output.

Syntax: LL file|lu (+LL:file|lu in runstring)
file File descriptor of list file. If file descriptor is specified, file must not exist or must have type extension MAP.
lu LU number of list device.

LO (List Program Attributes)

Purpose: List program attributes during relinking.

Syntax: LO

MA (Display Load Map)

Purpose: Display load map on terminal when running LINK interactively.

Syntax: +MA (in runstring only).

MS (Multiple Search)

Purpose: Search a library file.

Syntax: MS file

file File descriptor of file to be searched.

NA (Name)

Purpose: Set program file name.

Syntax: NA [file]

file File descriptor to be used for program file.

OR (Order EMA Area)

Purpose: Specify order in which EMA areas are allocated.

Syntax: OR area1 area2 area3 ...

arean Name of EMA area declared in source file.

OS (Operator Suspend)

Purpose: Suspend LINK (access to S=xx command? prompt allowed) until you enter the system GO command.

Syntax: OS

OU (Output)

Purpose: Specify program output file name. (LINK does not overwrite existing file by same name.)

Syntax: OU file

file File descriptor and subparameters for program output file.

PA (Page Align EMA Area)

Purpose: Start the next EMA area specified with an OR command on an even page boundary.

Syntax: PA

PR (Priority)

Purpose: Set the priority of the program.

Syntax: PR nn

nn Program priority, in range 1 to 32767 (default is 99).

PS (Page Align Overlays)

Purpose: Start overlays at page boundaries.

Syntax: PS

RC (Reverse Common)

Purpose: Reverses common usage. Background programs will have their common placed in the real-time common area. Real-time programs will have their common placed in the background common area.

Syntax: RC (+RC in runstring).

RE (Relocate)

Purpose: Include a relocatable file as part of current segment.

Syntax: RE file

file File descriptor of file to be included in program.

RM (Relocate Module)

Purpose: Include a module as part of current segment. The specified file is searched until a module is found that contains the specified symbol as an entry point. The module is then relocated.

Syntax: RM file symbol

file File descriptor of file containing module to be relocated.

symbol Entry point for module to be relocated.

RO (Reorder)

Purpose: Rearrange program modules to reduce number of base page links.

Syntax: RO (+RO in runstring).

RT (Real-time)

Purpose: Specify a real-time (type 2) program.

Syntax: RT (+RT in runstring).

SC (System Common)

Purpose: Specify that blank common referenced by program will be placed in blank system common.

Syntax: SC (+SC in runstring).

SE (Search)

Purpose: Search a library file to satisfy undefined external references.

Syntax: SE [file]

file File descriptor of library file. Default search is through snapshot and system library files.

SH (Shareable EMA)

- Purpose:* Specify that the declared EMA resides in the shareable EMA partition specified.
- Syntax:* SH label
- label Partition name (up to 6 characters).

SN (Snapshot)

- Purpose:* Define or display snapshot file. (Default snapshot file names are SNAP::0, /SYSTEM/SNAP,SNP, SNAP,SNP, or SNAP.6::0 unless changed by this command.)
- Syntax:* SN [file]
- file File descriptor of snapshot file. If this parameter is omitted, the snapshot file name is displayed.

SS (Subsystem Global Area)

- Purpose:* Specify use of subsystem global area.
- Syntax:* SS

SZ (Size)

- Purpose:* Specify number of physical memory pages required to run program.
- Syntax:* SZ pages|+pages (+SZ: in runstring)
- pages Number of pages, in range 2 to 32.
- +pages Number of pages, in range 2 to 32, to be added to pages required by program.

TR (Transfer)

- Purpose:* Transfer control to a file containing LINK commands.
- Syntax:* TR file
- file File descriptor of command file. If no type extension specified, LINK assumes .LOD.

VM (Virtual Memory Size)

Purpose: Specify that the program uses VMA; specify size of VMA backing store file.

Syntax: VM [pages]

pages Maximum number of pages in backing store file.
Default is 8192.

WS (Working Set Size of VMA)

Purpose: Specify working set size of VMA, excluding one-page PTE.

Syntax: WS [size]

size Number of pages, in range 2 to 1022 (default is 32).

* (Comment)

Purpose: Allows entry of comments; ignores remainder of line.

Syntax: * (in column one).

? (Help)

Purpose: Display help information.

Syntax: ? [command]

command Help on a particular command. Default is a summary of all commands.

Interactive Utilities

General System Utilities	
COMPL and CLOAD	7-1
EXT	7-2
FLAG	7-3
FPORT	7-4
KEYS (Define Key Functions)	7-5
KYDMP (Program Key Functions)	7-5
LGAT (Log Track-Assignment Table)	7-6
LUPRN (System Configuration Display)	7-6
MERGE	7-6
OLDRE	7-7
PRINT	7-7
SCOM (Source File Comparison)	7-8
SPORT (Serial Port Analyzer)	7-9
Backup Utilities	
FC (File Backup Utility)	7-9
FST (File Storage to Tape)	7-11
LIF (HP Computer Systems File Copy)	7-13
PSAVE (Physical Save)	7-14
PRSTR (Physical Restore)	7-15
PCOPY (Physical Disk-to-Disk Copy)	7-16
READT (Restore Disk Cartridge)	7-16
TF (Tape Filer)	7-17
WRITT (Save Disk Cartridge)	7-17
File System Utilities	
FOWN	7-18
FPACK	7-18
FREES	7-19
FSCON	7-19
FVERI	7-20
MPACK	7-20
Disk Formatting Utilities	
FORMC (CS/80 Maintenance Utility)	7-21
FORMT (ICD and MAC Maintenance Utility)	7-22
Help Lookup Utilities	
CALLM	7-23
CALLS	7-23
CMD (General Purpose Help)	7-24
GENIX (Create Indexed Sequential File)	7-24

Compilers	
FORTRAN Compiler	7-25
Macro Compiler	7-26
Pascal Compiler	7-27
Linking and Loading Utilities	
INDXR (Create an Indexed Library File)	7-28
LINDX	7-28
MLLDR/LOADR	7-29
MLLDR/LOADR Commands	7-30
MLLDR Segmentation Commands	7-32
SGMTR (Create MLLDR Command File)	7-32
SXREF (Verify MLLDR Command File)	7-32
Debugging Utilities	
DBUGR and MLSDB	7-33

COMPL and CLOAD

```
[RU,]COMPL,source[,list[,reloc[,compiler[,options]]]]  
or  
[RU,]CLOAD,source[,list[,reloc[,compiler[,options [ldcmd  
[loader]]]]]]
```

These utilities automatically invoke the appropriate compiler or assembler for a specified source file. CLOAD, in addition, schedules LOADR.

source	Name of source file.
list	Disk file, LU, or “-” for list file. “-” creates file ‘source for list file if source begins with &, or source.lst if source ends with valid type extension. For CLOAD list must be an LU (default = user’s terminal).
reloc	Name of file or “-” for relocatable code. “-” creates file %source for relocatable code if source begins with &, or source.rel if source ends with valid type extension (no default).
compiler	Compiler to use.
option	Optional file containing a control string option list (Pascal only) or an option string passed to all other compilers or assemblers.
ldcmd	Optional LINK or LOADR command file descriptor that contains the load commands.
loader	Loader to use (for example, LINK, LOADR).

EXT

[RU,]EXT[,-options],srcfile[,outfile]

- srcfile Source relocatable (type 5) file to be searched for external references.
- outfile Output file to receive list of externals found in srcfile. If outfile exists, EXT output is appended to it, unless R option is specified.
- options Any of the following:
- C Condense the output list: separate the listed externals with spaces, rather than <crLf>. Available only if outfile = terminal.
 - Lnn Lengthen output line to *nn* characters. Default = minimum = 80; maximum = 134. Force C option; available only if outfile = terminal.
 - Snn Scroll the output *nn* lines at a time (default = 22), prompt for more after each *nn* lines. Use -S0<terminalLU> for continuous printing without prompting. Available only if outfile = terminal.
 - N Name: include the **nam** record of the routine in which the externals are found.
 - T Identify entry point and external names.
 - V Verbose; combine C, N, and T options.
 - R Replace contents of outfile if it exists, rather than appending to it.
 - Efile Error message file (default = terminal). If file already exists, messages are appended.
 - Ffile Find only the externals in the named patterns file.
 - Ifile Ignore the externals in file.

FLAG

[RU,]FLAG,pfile[,–options],sfile[,...]

pfile Patterns file; SEP.6 or custom patterns file.

sfile Source file or files to be searched.

–options Any combination of the following:

C Count: for each word matched, print a count of lines that contain a match.

K Make case significant in determining a match. By default, case is not significant ('a' matches 'a' or 'A').

M Print source file name before each output line (default when more than one source file is specified).

N Print the line number of each matched line before each output line.

V Verbose; print all lines in the file with line numbers. Flag matches in lines and print count for each word.

P Pascal

B BASIC

F FORTRAN

A Assembler

Language options assume that all files specified after a language option are source code in that language, all comments in the source are ignored.

Ofile

Specifies output file (default = terminal). The file name must follow the option letter with no space.

FPORT

[RU,]FPORT, -E, tmap[, tdev]

or

[RU,]FPORT, -I[FM], tmap[, tdev]

- E Specify Export mode.
- I Specify Import mode.
- tmap The transport map file defining the files to be exported/imported.
- tdev The device LU of the external medium through which **tmap** and the transported files are copied. The default is to LU 8.

Mutually exclusive flags, applicable in Import mode only:

- F Force the use of tmap as the transport map and ignore the transport map in the transport file.
- M Map only. Import only the transport map from the transport file into **tmap**.

Transport Map Flags:

- F Treat the export name as a FMGR name.
- f Treat the import name as a FMGR name.
- b Treat the file as a binary file. This flag has meaning only for HP-UX files.

KEYS (Define Key Functions)

[RU,]KEYS[,console[,list]][-S]

- console LU of the display terminal to receive the KEYS prompts. The default is to LU 1 (your terminal).
- list Device on which the command set file is to be listed. The default is to your terminal.
- S Inhibit screen display of soft keys definitions.

KEYS command summary:

- F [file] Load current command set from file to be modified. If file is not given, reports current file name.
- M [key#] Modify softkey definition for specified key number.
- L List current softkey definition.
- W [file] Write current sofkey definitions to file. If file is not given, last file name used is replaced.
- V Redraw softkey screen display.
- ? Help.
- EX Exit.

KYDMP (Program Key Functions)

[RU,]KYDMP,[options,][console,]keyfile

- options The options are as follows:
 - L Do not dump labels, just program the softkeys.
 - M Do not memory-lock labels at the screen top.
- console LU of the display terminal to receive the KEYS prompts. The default is to LU 1 (your terminal).
- keyfile Name of the KEYS file containing the command set.

LGTAT (Log Track-Assignment Table)

[RU,]LGTAT[,lu[,form]]

- lu LU to which LGTAT will direct output. Default is the log LU.
- form Output format. The format is specified as:
 - 0 = abbreviated form (default)
 - 1 = complete form

LUPRN (System Configuration Display)

[RU,]LUPRN[,lu[,AL[,LU[,SC[,TY[,DV[,??]]]]]]]]

- lu LU number of the list device (default is user terminal).
- AL List all devices, sort by system LU.
- LU[:n:n] List all devices within specified range (optional), sort by system LU.
- SC[:n:n] List all devices with assigned select codes within specified range (optional), sort by select code.
- TY[:nB:nB] List all devices within specified octal range (optional), sort by device type.
- DV List system driver table, followed by AL option listing.
- ?? List descriptive summary of LUPRN and runstring options.

MERGE

[RU,]MERGE [-options] [fromfile fromfile . . .] [destfile]

Combine two or more input files into a single output file.

- options The available options are:
 - L Suppress listing of file names.
 - D Remove debug records.
 - O Overwrite existing file without user confirmation.
 - V Verify existing output file to be overwritten.
 - Z Suppresses zero-length records between files.
- fromfile Input file(s) to be merged; fromfile may be a command file containing a list of files to be merged with type extension .MRG, a mask of files, or an interactive LU.
- destfile File or LU receiving merged files.

OLDRE

[RU,]OLDRE,filesdes

Change extended relocatable record formats to nonextendable formats.

filesdes Name of type 5 file.

PRINT

[RU,] PRINT [file1[file2[...[lu][options]]]]

Print one or more files to a printer or a file. If entered without parameters, the status of jobs in the printer queue is displayed.

PRINT options can be placed anywhere in the runstring provided a plus sign precedes each option.

file1...	Name of file or file mask for files to be spooled for printing.
lu	LU of printer. Default = 6.
options	One of the following:
+A:file	Append output to the specified file.
+B:string	Print the specified string as banner headline.
+C:[ON OFF]	Set the carriage control option on or off.
+F:number	Advance paper the specified number of pages after all files are printed.
+I:number	Indent the specified number of pages.
+M:number	Merge consecutive files with the specified number of blank lines between the files.
+N	Produce a numbered listing for each file in the runstring.
+O:file	Send each file listed in the runstring to the specified file.
+P[:OK]	Purge printed files that have been sent to the line printer.
+Q	Suppress file mask verification.
+S	Suppress message "Print job supervised by PRINX".
+X:number	Print all files contained in the runstring the specified number of times.
+W:dir	Search specified directory, not current working directory.
+?	Return a short explanation of available options.

SCOM (Source File Comparison)

[RU,]SCOM,file1,file2,[[+ | ~]listfl][options],[rematchlns]
,[maxchars],[difflimit]

- file1,file2 Input files to be compared.
- listfl Listing file (default = user's terminal). + appends listfl to existing file, ~ overlays an existing file.
- option Any of the following, concatenated without intervening commas:
- F1 Report lines unique to file1 (default).
 - F2 Report lines unique to file2 (default).
 - BO Report lines common to both files.
 - NN Suppress line numbers on report.
 - NH No heading on listing (parameter information).
 - NT No trailer on listing (number of mismatches).
 - TB Trailing blanks are significant.
 - IB Trailing blanks are insignificant.
 - D<x> Wildcard; <x> matches any character in the other file.
 - C<x> Ignore lines with <x> in column 1 but otherwise blank when rematching.
 - IT Ignore Edit/1000 time stamps.
 - IC Ignore compile times in relocatable records for binary record compares.
 - ET Create an Edit/1000 transfer file to convert file1 to file2.
 - ER Same as ET but add an Edit "ER" command to end of transfer file.
 - BR Binary record-by-record compare.
 - BB Binary block-by-block compare.
 - TC Force text compare on binary files.
- rematchlns Number of consecutive lines which must match in the input files before a mismatch is ended (default = 3).
- maxchars Maximum record size in the input files (default = 256 characters).
- difflimit Maximum number of differences to report; all differences reported if difflimit is not specified.

SPORT (Serial Port Analyzer)

[ru,]SPORT [port_lu [,display_lu [,waitflag]]]

- port_lu Serial port LU for which the current status is requested. Default is the session LU.
- display_lu LU to which the output is directed. Output can be directed to another LU, but not to a file. Default is the session LU.
- waitflag Allows you to specify a non-zero value to force a “wait” for a busy port; SPORT “hangs” on the port until the current request completes.

FC (File Backup Utility)

[RU,]FC[,command string]

command string

A string of FC commands. If defaulted, commands are entered interactively from the user’s terminal.

FC commands:

AB Abort FC, including any active group copy.

AG Halt the GR command.

CF,namr
 Establish title for tape comment file.

CL[, -tlu][,K]
 List cartridges you can access.

CLAL
 List all cartridges in the system.

CO[,srce[,dest[,optns[,file1[,file2[,msc]]]]]]]
 Copy files as specified by parameters.

*comment
 Enter comment in command file.

DE[,srce[,dest[,optns]]]
 Set default source, destination, and options for subsequent CO commands.

DL[,srce[,msc[,optns]]]
 List the directory list.

- EC[,ON|OFF]
Turn ON or OFF echoing of commands to list device. Default is ON.
- EG End of the group copy commands.
- EX Exit FC.
- GR Execute the following CO commands as a single operation.
- HE[,key[,lu]]
Display FMGR error description.
- LC,-tlu[,K]
List the comment file.
- LH,-tlu[,K]
List the header file from tape.
- LL,namr|-
Set list device or file. '-' resets list device to terminal.
- SC,cart
Specify disk cartridge for internal scratch files.
- TI,title
Establish title for tape header file.
- TR Transfer from FC command file.
- TR,namr|-
Transfer to FC command file. '-' transfers control to terminal.
- ? Display a summary of available commands.

FST (File Storage to Tape)

[RU] FST [cmd | cmd | ...] [option]...[option] [ON|OFF]

cmd one or more of the following FST commands separated by a vertical bar (|). If not given, FST goes into interactive mode.

Information Commands:

HE|? [cmd] Display command help information.
SH Show user-selected states of FST.

Backup and Restore Commands:

BA mask [dest] [security] Select files to back up from disk.
DF file_desc Specify directory file name, location or size.
GO Begin data transfer to/from archive.
RE [mask] [destmask] [GR|EG|AG] Select files to restore to disk.
SC file Select comment file for the archive.
TA [ON|OFF|A|B|C] Use UNIX TAR format. A=ASCII, B=binary, C=case sensitive.
TI title Specify a title for the archive header.
UN [mask] Remove files from selection in directory file.

Listing Commands:

DL [mask] List directory of files on the archive.
LC List comment file of the archive.
LH List the archive header.
LI [mask] List files selected for backup/restore.
LL device|file [A] [O] Change or select a log device/file. 'A' appends to and 'O' overwrites log file.
LN [mask] List the non-selected files of the directory file.

Tape Related Commands:

MT [lu|file] Select tape LU or archive file name.
NE [append#] Advance to the next append of data.
PO [append#] Position to a specific append of data.
PR [append#] Rewind to previous append of data.
SD [density] Set the tape density.
SE Secure (lock) tape LU or (open) archive file.

Miscellaneous Commands:

EX	Exit FST.
TR filename	Transfer control to a command file.
RU program	Run a program.
*	Comment.

/[n] Display FST command stack.

option one or more of the following options:

A	Append to data already on tape.
B	Only show errors and status messages.
C	Clear the disk file's backup bits.
D	Replace duplicate files.
F	Restore files from a partially overwritten tape.
I	Inhibit tape rewind between backups.
K	Keep tape online.
L	Lock any disk LUs used.
M	Minimize FST directory file size during restores.
N	Back up symbolic links as normal files (RTE-A only).
O	Restore files to their original main size.
P	Purge disk files after backing up the files.
Q	Report messages only to the log device/file.
R	Rewind and go offline at exit.
S	Search through appends during restore.
U	Replace duplicate file if file has been updated.
V	Verify the files during the backup/restore.
W	Back up all the blocks reserved for the file.
Y	Write over the tape without asking.
Z	Pause during restore phase for disk full errors.

LIF (HP Computer Systems File Copy)

[RU,] LIF [,cmdfile[,listfile]]

- cmdfile Command file containing LIF commands, must be an FMP file. Default is the display terminal.
- listfile LU or FMP file to which error messages or file/device listings are reported. Default is the display terminal.

LIF Commands:

- CO, -lu, destmask
Copy all files from LIF medium onto RTE disk cartridge.
- DL[,mask[,level]]
Display files mounted on LIF medium.
- EX (or EN, /E)
Terminate LIF.
- HE (or ?, ??)
Display all LIF commands.
- IN, lu[,vol label[,dir start[,dir length]]]
Write volume label and blank directory on LIF medium.
- LI, file
Copy files, either FMP or LIF, to list file/device.
- LL[,FMPfile]
Change list file or device.
- MC, lu
Mount LIF cartridge.
- PK Pack files on LIF medium.
- PU, LIF file desc
Remove file from LIF medium.
- RN, oldLIFfile, newLIFfile
Change name of LIF file to a new one.
- ST, srcfile[,destfile]
Create files from existing files on FMP/LIF medium to FMP/LIF medium.
- SV[,severity level]
Modify amount of error checking done.
- TR[,FMP file desc]
Transfer control from one RTE file or LU to another. Default is the scheduling LU.

PSAVE (Physical Save)

```
[RU,]PSAVE[,input[,srceLU[,destLU[,file#[,opts[,hcpy  
[,tape density[,title]]]]]]]]]
```

- input** LU (or file) from which PSAVE parameter inputs are to be read. If input is from a file or a non-interactive device, no other parameters may be specified. Default is your log device.
- srceLU** LU of disk subchannel to be saved. If UN option is specified, srceLU is a target to the unit and may be any LU on the disk. If MU is specified, this parameter is meaningless and is ignored. In any other case, this is a required parameter.
- destLU** LU of tape transport or tape cartridge to receive the saved data. Under session monitor, the LU specified must be in your Session Switch Table (SST). Default is LU 8.
- file#** Number to be assigned to the saved file. Specifies start location on the tape for the save. Default is current tape position for tape transports or file #1 for CS/80 cartridges.
- opts** Any of the following options. If no options are specified, the default is to LU save without verify.
- VE Turn on verify option.
 - LU Save is to be an LU save.
 - UN Save is to be a disk unit save. srceLU can be any LU on the disk unit.
 - PB Unit save in CS/80 pushbutton-restorable format.
 - MU Multiple LU saves in one pass.
- hcpy** LU of the device on which information about the save is to be printed as a record of the save operation. Default is your log device.
- tape density** Must be 800, 1600, or 6250. Default is to use the last tape density specified.
- title** Title (to a maximum of 40 ASCII characters) that will be placed in the tape header.

PRSTR (Physical Restore)

[RU,]PRSTR[,input[,dest LU[,srce LU[,file#[,opts[,hcpy]]]]]]

- input** LU (or file) from which parameter inputs are to be read. If the input is from a file or a non-interactive device, no other parameters may be specified. Default is log device.
- dest LU** LU of the disk subchannel on which data is to be restored. If UN option is specified, the LU is a target and may be any LU on the disk. If SE is specified, this parameter must be omitted. In all other cases, the parameter is required.
- srce LU** LU of the tape transport or CS/80 cartridge from which the save file is to be read. Under session monitor, the LU specified must be in your Session Switch Table (SST). Default is LU 8.
- file#** Number of the file to be read. Default is the current tape position for tape transports, or file #1 for CS/80 cartridges.
- opts** Any of the following option. If no options are specified, the default is either LU restore with no VErify if the save was an LU save, or UN if the save was a unit save.
- VE** Turn on data verify option.
 - DE** Do not prompt for verification of tape file header. Do not prompt for OK TO CONTINUE?.
 - UN** Unit restore. This is the default if the tape is a unit save or a from-to save.
 - LU** LU restore. This is the default if the tape is an LU or MU save.
 - SE** Selective restore from a unit save, or multiple LU restores in one pass. If this option is selected, dest LU must be omitted; PSAVE will prompt for file:LU pairs. Not valid for unit save in pushbutton-restorable format or unit save in formats other than those created by PSAVE. All LUs must be of the same disk class.
 - PB** Unit save in CS/80 pushbutton-restorable format.
- hcpy** LU of the device on which information about the restore is to be printed. Default is your log device.

PCOPY (Physical Disk-to-Disk Copy)

[RU,]PCOPY,[input],srce LU,dest LU[,VE[,hcpy]]

input	LU (or file) from which the parameter inputs are to be read. If input is from a file or a non-interactive device, no other parameters may be specified in the runstring. Default is the log device.
srce LU	LU of the disk subchannel to be copied.
dest LU	LU of the disk subchannel to which the data is to be copied.
VE	Verify option. PCOPY spares the destination subchannel tracks in a prepass over the disk before data is written. Default is to suppress verify.
hcpy	LU of the device on which the read or verify errors are logged. Default is to the log device.

READT (Restore Disk Cartridge)

[RU,]READT[,disk[,MT:lu[,type[,SI:nnn]]]][,IH][,VE |CO]

disk	The negative logical unit (LU) number or Cartridge Reference Number (CRN) of the cartridge to which the previously saved cartridge is to be restored.
MT:lu	The LU number of the mag tape unit (default is LU 8). Either a positive or negative LU can be specified.
type	Designate the type of cartridge to be restored: P for private cartridge, G for group cartridge.
SI:nnn	The desired size of the cartridge to which the mag tape contents are to be restored. The size is specified in number of tracks (default is the size of the cartridge saved on the mag tape).
IH	Inhibit tape rewind (default is to rewind).
VE	Verify data transfer. Either VE or CO may be included, but not both.
CO	Perform word-by-word comparison of tape to restored cartridge.

TF (Tape Filer)

[RU,]TF[command]

Back up and restore files from the file addressing space to magnetic or CS/80 cartridge tape.

command Default = interactive mode.

?	List commands and their syntax.
CO file	Copy files as specified by parameters.
TI title	Set title for tape header file.
DE filedesc	Set default source, destination, and options for subsequent COPY commands.
GR,EG,AG	Group multiple COPY commands into a single COPY operation.
LL file	Set list device or file for LH or DL.
LH lu [K]	List header file from DF tape.
DL mask	Compile directory list of TF tape.
TR file	Transfer to or return from TF command file.
EX	Exit TF.
*	Comment.

WRITT (Save Disk Cartridge)

[RU,]WRITT[,disk[,MT:lu][,IH][,DC][,VE][...]

disk	The negative logical unit (LU) number or the Cartridge Reference Number (CRN) of the disk cartridge to be saved on mag tape.
MT:lu	The LU number of the mag tape unit (default is LU 8). Either a positive or negative LU can be specified.
IH	Inhibit tape rewind (default is to rewind).
DC	Disable overlay check.
VE	Verify data transfer.
"..."	Comment to be appended to tape header; 40 characters maximum.

FOWN

[RU,]FOWN[mask]

Display owner of and disk space used by files specified.

mask File mask. Default = all CI files.

FPACK

[RU,]FPACK lu

Rearrange files on file system volume, increasing largest free space on volume.

lu LU of volume to be packed.

FREES

[RU,]FREES [options] [DiskLu] [DiskLu:DiskLu] ...

Report total free space and size of largest free space on CI file system volume.

options one or more of the following:

-v Inhibit the inverse video bar graph.

+l:lu Send the listing to the given LU.

-g Bar graphs are not relative to the largest disk.

*disk*sz where *disk*sz is one of the following:

+t Report disk size in number of tracks.

+m Report disk size in Mbytes.

sort where *sort* is one of the following:

-s Inhibit sort, report in cartridge list order.

+h Sort by largest hole size.

+d Sort by disk size.

quiet where *quiet* indicates quiet mode and is one of the following:

+q return all free space information to the scheduling program in \$return_s.

+qd return only disk size in \$return_s.

+qf return only free size in \$return_s.

+qh return only largest hole size in \$return_s.

+q_r return only reserved size in \$return_s.

+q%f return only percent free in \$return_s.

+q%m return only percent max in \$return_s.

DiskLu Disk LU to display; it can be repeated.

DiskLu:DiskLu

Disk LUs to display; if no disk LU is specified, all CI volumes are listed.

FSCON

[RU,]FSCON lu

Convert FMGR cartridge to CI structure.

lu LU of FMGR cartridge to be converted.

FVERI

[RU,]FVERI[lu|mask][options]

Verify that data within a hierarchical file system volume is consistent. Fix some of the inconsistencies found by the verification.

lu	LU of volume to be verified. Default = all volumes.
mask	Mask describing a directory in which all files and/or subdirectories are to be verified.
options	One or more of the following, in any order: +L,file lu List file or device for errors. +B Verify bit map only; illegal if a mask is also specified. +FB Fix the bit map. +FD Fix illegal directory entries (set purged). +FF Fix file directory information. +OK Perform fixes without asking each time. ?? Online help.

MPACK

[RU,]MPACK[options] mask [options]

Compact CI files and pack CI file volumes.

options	MPACK has compacting, reporting, packing, and logging options. Precede options with "+". Enter options anywhere in the runstring but enter them only once. Compacting takes place before packing.
---------	---

Compacting/Reporting Options:

- +R Remove extents from files.
- +T[n[%]] Truncate all wasted blocks from files, or truncate all but *n* or *n%* of wasted blocks.
- +C[n[%]] Remove all extents and truncate all wasted blocks or truncate all but *n* or *n%* of wasted blocks; +C cannot be used in same the runstring with either +T or +R.
- +En Select files with *n* or more extents.
- +Wn Select files with *n* or more wasted blocks.
- +Wn% Select files with *n%* or more wasted blocks.
- +A AND the +E and +W qualifiers.
- +D Include directories in the compacting.
- +Q Quiet mode. Suppress individual file information and only report totals.

Packing Options:

- +P Pack disk LU.
- +OK OK to overlay data during pack.

Logging Option:

- +L Log output to a file or device.

FORMC (CS/80 Maintenance Utility)

[RU,]FORMC[:IH],[list LU],option,lu[,opt_parms]

- :IH Inhibit automatic renaming of the program. :IH must be specified when Format or SPare is selected since these options require FORMC run without renaming.
- list LU LU of list device to which FORMC will direct messages.
- option One of the following FORMC commands:
 - HE (or ?) Display FORMC commands.
 - FO Format disk volume or cartridge tape.
 - SP Spare defective blocks.
 - VE Check integrity of disk LU or CTD tape.
 - AB (or EN, EX, /E) Terminate FORMC.
- lu LU of the CS/80 disk or tape to be accessed by FORMC for the specified operation.
- opt_parms Parameters applicable to the specified option.

FORMT (ICD and MAC Maintenance Utility)

[RU,]FORMT[,input[,cmd,disk_lu[,n]]]

- | | |
|---------|--|
| input | LU of the device from which the FORMT parameters are to be read. Default is the log device. |
| cmd | One of the following FORMT function commands:
FO Format floppy disk.
IN Initialize specified disk.
RE Reformat specified disk LU.
VE Verify data on specified disk LU. |
| disk_lu | Disk LU; positive integer between 0 and 256. |
| n | Number of “fill” sectors (interleave factor). |

CALLM

[RU,]CALLM [-options] cmdfile destfile

- options String of one or more of the following characters preceded by a dash:
- l suppress listing the names of files read.
 - o overlay an existing destfile.
 - v verify that an existing destfile should be overlaid.
 - c inhibit text compression of destfile. Default is to compress the text in destfile.
- cmdfile File containing a list of text files to be read, one per line. The CALLS input is extracted from each of these files and merged into destfile.
- destfile Destination file to be used as an input file for CALLS. If compression is performed, this file will be of type 6004.

CALLS

[RU,]CALLS [-flags] [keyword]

- flags One or more of the following flags preceded by a dash:
- C catalog The name of the CALLS catalog to use. Default directory “/catalogs” and type extension “.call” are added to given name. Default catalog is “/catalogs/calls.call”.
 - L listfile Divert text listing to the named file. By default, the text is listed to the terminal.
 - P pagesize Set number of lines per page for “More...” prompting on the terminal. Default is 22 lines.
 - B Build index file and terminate.
- keyword Keyword for which the associated text is to be listed. If not given, default keyword (“[default]”) for the selected catalog is listed.

CMD (General Purpose Help)

[RU,]CMD[,key[,listfile[,helpfile[,NI|SK]]]]

key	Used in searching the indexed file for a match (1-24 characters in length).
listfile	File or LU to receive output (default = user's terminal).
helpfile	Name of message file to be searched.
NI	Non-interactive mode.
SK	Single key mode.

GENIX (Create Indexed Sequential File)

[RU,]GENIX,input[,list],output

input	Input text file of keys and text to be indexed.
list	Output list file name (default = log LU).
output	Type 1 indexed file created by GENIX, searched by the CMD utility.

FORTRAN Compiler

[RU,]FTN7X,source[,list[,relocatable[,pg[,opt[,directive]]]]]

- source Disk file or LU for source file.
- list Disk file, LU, or “-” for the listing. “-” creates file 'source for listing if source begins with & (default = user's terminal).
- relocatable Name of file or “-” for relocatable code. “-” creates file %source for relocatable code if source begins with & (no default).
- pg Line count per page.
- opt Optional control statement that overrides the source file control statement. Options are as follows:
- B Override any options in the control statement (except I, J, X, Y, and E).
 - C Output a cross-reference symbol table listing to the list file.
 - D Compile debug lines.
 - E Select EMA transparency mode; causes arguments to be passed using 32-bit addresses.
 - I Integers are stored in one word (default).
 - J Integers are stored in two words.
 - L Output source to list file.
 - M Output a mixed listing of both the source and the object program to the list file.
 - n* Error routine *n* supplied. *n* is a decimal digit 1-9 which specifies an error routine ERR*n*.
 - P Enable warning about possible overlap in character assignments.
 - Q Include the approximate relocatable address of each statement on the listing.
 - S Generate symbolic debug records.
 - T Output symbol table for each main or subprogram to the list file.
 - X Double precision is stored in three words (default).
 - Y Double precision is stored in four words.
- directive compiler directive beginning with a dollar sign (\$).

Macro Compiler

```
[RU,]MACRO,source[,list[,dest [ ,&.RS1=string1
                                ,&.RS2=string2
                                ,WORK=filedescriptor
                                ,OPT=opt
                                ,LINES=pg
                                ,opt
                                ,pg ]]]]
```

- source Disk file or LU for source file.
- list Disk file, LU, or “-” for the listing. “-” creates file 'source for listing if source begins with & (default = user's terminal).
- dest File name or LU to which binary output is directed.
- string1, string2
Character values that are assigned to global system assembly time variables of the same names in the source program.
- filedescriptor
File path to be used by the Macroassembler for scratch files.
- pg Line count per page.
- opt Optional control statement that overrides the source file control statement. Options are as follows:
 - A Absolute assembly, the addresses generated by the Macroassembler are interpreted as absolute locations in memory.
 - C Output a cross-reference symbol table to the list file.
 - F The floating point machine instructions are to be used instead of the software simulation routines for: FIX,FLT,FDV,FMP,FAD,FSB.
 - I Generate microcode instructions when possible.
 - L Output source listing to list file. This includes both the opcode and the address of the operand if it is a memory reference instruction.
 - M Create macro library
 - N,Z Selective assembly, sections of the program are to be included or excluded at assembly time depending upon the option specified.
 - O Generate old records (compatible with RTE-IVB and earlier operating systems).

- Q Output source listing to list file. This includes only the operand address for single-word memory reference instructions, otherwise the entire object code is listed.
- R Relocatable assembly, the object program may be loaded anywhere in memory.
- S Generate symbolic debug records.
- T Output symbol table to list file.
- X No EAU hardware on machine.
- +DC=<yymmdd
Install software date code.
- +SF=filedescriptor
Put file descriptor in NAM record instead of source file.

Pascal Compiler

[RU,]PASCAL,source[,list[,relocatable[,option]]]

- source Disk file for source file (default = user's terminal).
- list Disk file, LU, or "-" for list. "-" creates file source for listing if source begins with & (default = user's terminal).
- relocatable Name of file or "-" for relocatable code. "-" creates file %source for relocatable code if source begins with & (no default).
- option Optional file containing the number of pages of EMA used by the compiler (default = 15). If option=1, enter option at user's terminal.

INDXR (Create an Indexed Library File)

[RU,]INDXR[,cmdfile]

cmdfile Command file name or input device LU
 (default = user's terminal).

LINDX

[RU,]LINDX,inputFile,outputFile[, +NL]

or

[RU,]LINDX,SYSTEM,snapfile[,library...][, +NL]

or

[RU,]LINDX,SYSTEM,snapfile[, +LI libList][, +NL]

Index library files for faster searching.

inputFile File to be indexed.

outputFile Indexed library (different from inputFile).

+NL Inhibit listing of entry points.

snapfile Resulting snapfile for the system.

library... Library to include in the snapfile. Up to 10 separated by
 commas.

libList File name containing a list of library file names.

MLLDR/LOADR

[RU,] LOADR
MLLDR [command[,input[,list[,opcode[,format[,partn[,size
[,profile]]]]]]]]]

- command A command file namr, or input device LU
(default = user's terminal or LU 5 if batch).
- input The file name of the relocatable main program or the
LU of the relocatable input (no default).
- list List LU, or file name namr. If a file name is specified,
the file must not already exist unless it begins with (').
(Default = user's terminal or LU 5 if batch).
- opcode Default = BGNCTE (LOADR) or EBNCTE (MLLDR)
- BG Background program.
 - RT Real-time program.
 - LB Large background program.
 - EB Extended background program.
 - SC System COMMON.
 - RC Reverse COMMON.
 - NC No COMMON.
 - SS Use subsystem global (SSGA).
 - PE Permanent program.
 - TE Temporary program.
 - RP Replace permanent program (do not also specify
PE).
 - VM Virtual memory program (do not also specify PE).
 - EM EMA program (used in MLS programs).

format	DB Append DBUGR (LOADR) or MLSDB (MLLDR) subroutine to the program.
	LE List entry points and base page links.
	NL No listing desired.
	DC Don't copy. Multiple copies of the program are not desired.
	MP Use current page links, except for external references.
	CP Use current page links, including external references.
	BP Use base page links only (default).
partn	The specific partition number in which program is to be executed.
size	Allows a logical address space larger than the program size. Permits use of a dynamic buffer at the end of the program.
profile	LU for profile output. Default = no profile is done (MLLDR only).

MLLDR/LOADR Commands

SE	Search the system disk library for undefined externals.
SE,file	Search specified file for undefined externals.
MS,file	Search specified file for undefined externals. The file is searched multiple times to satisfy backward references.
RE,file	Load specified file, which may be a program, subroutine, or segment.
LO,xxxxxB or +n	Change the load address of the next module to be relocated to the specified address or offset <i>n</i> pages.
LI,file	Set up file as a library file. Up to 10 files may be specified.
SL	Search all files specified in the library command.
TR,file	Go to file for succeeding loader commands.
TR	Return to command file suspended when the undefined external was encountered.

FO	Force load a program or segment.
DI	Print list of undefined externals.
EC	Echo input commands on list device. ** (See note below)
EN, EX, or /E	End of command input.
AB or /A	Abort the loader immediately.
*	Comment line.
AS,xx	Assign the relocated program to partition xx. **
SZ, yy, or +n	Allows a logical address space larger than the program size. Permits the use of a dynamic buffer at the end of the program. **
LL,namr	LU or file name for listing. If a file, it must not already exist, unless its name begins with (?). **
OP,opcode	Specify an opcode parameter. See Opcode section of MLLDR/LOADR Operation. **
FM,format	Specify a format parameter. See Format section of MLLDR/LOADR Operation. **
PF,lu	Append profiling subroutine to the program. List profile output to LU (MLLDR only). **
VS,xxxxxx	Assign VMA size of (xxxxx+1) pages to the VMA program. **
WS,yyyy	Assign working set size of yyyy pages to the VMA program. **
SH,label	EMA area of program is to reside in the specified shareable EMA partition. **
SA,xx	Reserve xx words of local SAVE area for FORTRAN. **

** Specification of the these commands must precede specification of any RELOCATE or SEARCH command.

MLLDR Segmentation Commands

- M.x.y** Place the following routines (until the next M or D command) in the specified memory-resident node (default = main).
- D.x.y** Place the following routines (until the next M or D command) in the specified disk-resident node (default = main).
- NA,name** Load the specified routine with the node being currently loaded. The routine must be found in a user specified library.
- SY,name** Load the specified routine with the node being currently loaded. The routine must be found in the system library.

SGMTR (Create MLLDR Command File)

[RU,]SGMTR,input[,output[,size[,main[,segoptions
[,loadoptions]]]]]

- input** Program's merged relocatable file.
- output** Name of the loader command file to be created (default = user's terminal).
- size** Maximum pages allowed in a path (default = 32).
- main** Main entry point of the program (default = first entry point in the input file).
- segoptions** "M" or "D" for memory or disk-resident nodes (default = M), or "A" option will cause an NA or SY command to be generated for every module in the program.
- loadoptions**
Loader opcodes EM, VM, PF or DB can be specified here.

SXREF (Verify MLLDR Command File)

[RU,]SXREF,command[,list]

- command** MLLDR command file name.
- list** Output list file name. If specified, a cross-reference listing will be provided.

DBUGR and MLSDB

DBUGR/MLSDB Command Conventions:

<code>\</code>	Escape (character mode) or backslash (block mode).
<code>/</code>	Forward slash.
<code>[]</code>	Input control character.
<code>LF</code>	Linefeed.
<code>CR</code>	Carriage return.

Breakpoint and Program Control

<code>n\B</code>	Set a breakpoint at location <i>n</i> .
<code>\B</code>	List breakpoint table and enter remove breakpoint mode.
<code>\\B</code>	Remove all breakpoints.
<code><seg>\B</code>	Set a breakpoint at entry to segment (DBUGR only).
<code><path>\B</code>	Set a breakpoint at entry to path (MLSDB only).
<code>n<seg>\B</code>	Set a breakpoint in segment at location <i>n</i> (DBUGR only).
<code>n<path>\B</code>	Set a breakpoint in the specified path at location <i>n</i> (MLSDB only).
<code>[”A]\B</code>	Break at entry to all segments or paths.
<code>[”N]\B</code>	Do not break at entry to all segments or paths.
<code>\P</code>	Proceed with program execution after a break trap.
<code>\\P</code>	Proceed with conditional breakpoint invoked.
<code>nP</code>	Proceed; do not trap until <i>n</i> breakpoints from now.
<code>n\P</code>	Proceed; do not trap until <i>n</i> breakpoints, including conditional breakpoints.
<code>nG</code>	Continue execution at location <i>n</i> .
<code>nX</code>	Execute the instruction <i>n</i> , then return control to DBUGR or MLSDB.
<code>\T</code>	Trace one instruction.
<code>n\T</code>	Trace <i>n</i> instructions.
<code>\\T</code>	Trace an entire subroutine call with no argument list or alternate returns.

Memory Examination and Modification

<i>n</i> < <i>s</i>	Define the symbol <i>S</i> to have the value <i>n</i> .
<i>n</i> /	Print and open location <i>n</i> .
[LF]	Print and open the next location.
[CR]	Close current location.
/	Open and print the contents of the location pointed to by the last quantity typed.
∨	Open and print the contents of the location pointed to by the last quantity typed, only looking at bits 0-10.
\\[TAB] or \\[CNTL I]	Open, set the location counter to, and print the contents of the location pointed to by the quantity typed, looking only at bits 0-10.
<i>m</i> [CR]	Change the contents of location <i>n</i> to <i>m</i> .
<i>m</i> [LF]	Same as <i>m</i> [CR] above and also print and open the next location.

Special Registers

\\M	Display contents of the special registers.
AREG/	Examine and modify A-Register.
BREG/	Examine and modify B-Register.
XREG/	Examine and modify X-Register.
YREG/	Examine and modify Y-Register.
EOREG/	Examine and modify EO Register.
MASK/	Examine and modify Search Mask.
CBVAL/	Examine and modify conditional breakpoint value.
CBMASK/	Examine and modify conditional breakpoint mask.
CBADDR/	Examine and modify conditional breakpoint address.
CBTEST/	Examine and modify conditional breakpoint test.
WRTL/	EXEC control word for DBUGR or MLSDB output device.
BRFLG/	Break flag; 0 = check for break, 1 = no check.

Map Examination Special Mode

<code>\J</code>	Put DBUGR or MLSDB into special mode.
<code>UM</code>	Display the user map.
<code>SM</code>	Display the system map.
<code>XL</code>	Set up a cross load from an address in the alternate map.
<code>PA</code>	Display the port A map.
<code>PB</code>	Display the port B map.
<code>A</code>	Abort the special mode with no change.

Print Mode Control

<code>\S</code>	Set print mode to symbolic instruction (default).
<code>\!</code>	Set print mode to symbolic instruction until [CR] is entered.
<code>!</code>	Print the last quantity typed as a symbolic instruction.
<code>\C</code>	Set print mode to constant.
<code>\=</code>	Set print mode to constant until [CR] is entered.
<code>=</code>	Print the last quantity typed as a constant.
<code>\H</code>	Set print mode to ASCII characters.
<code>\'</code>	Set print mode to ASCII characters until [CR] is entered.
<code>'</code>	Print the last quantity typed as two ASCII characters.
<code>\A</code>	Set print mode to address.
<code>\</code>	Set print mode to address until [CR] is entered.
<code><-</code>	Print the last quantity typed as an address.
<code>n/R</code>	Change the output radix to <i>n</i> .

EXEC Calls

EXEC 1, 2, 17, 18, 20 (I/O, READ/WRITE)	8-1
EXEC 21 (I/O, Class Get)	8-2
EXEC 3, 19 (I/O Control)	8-3
EXEC 6 (Program Completion)	8-5
EXEC 7 (Program Suspend)	8-5
EXEC 22 (Program Swap Control)	8-5
EXEC 8, 9, 10, 23, 24 (Program Schedule)	8-6
EXEC 14 (String Passage)	8-6
EXEC 13 (Status, Device)	8-7
EXEC 25 (Status, Partition)	8-7
EXEC 26 (Memory Size)	8-8
EXEC 11 (Time Request)	8-8
EXEC 12 (Timed Execution (Absolute Start))	8-9
EXEC 12 (Timed Execution (Initial Offset))	8-9
EXEC 4, 15 (Track Allocation)	8-10
EXEC 5, 16 (Track Release)	8-10
CLRQ (Class Ownership Management)	8-11
LURQ (Logical Unit Lock Program Call)	8-11
REIO, XREIO (Reentrant I/O)	8-12
RNRQ (Resource Management)	8-12
XLUEX (Extended LU EXEC Call)	8-13

EXEC 1, 2, 17, 18, 20 (I/O, READ/WRITE)

CALL EXEC(ICODE,ICNWD,IBFR,ILEN[IPRM1][,IPRM2],ICLAS)

ICODE	1	READ
	2	WRITE
	17	Class READ
	18	Class WRITE
	20	Class WRITE/READ

ICNWD Control word. If Z bit (12) is set, an additional control buffer specified by IPRM1 and IPRM2 is passed to the driver or to the program doing the GET call.

Bit	15	Not used.
	14	UB Do not buffer.
	13	Not used.
	12	Z IPRM1 is buffer of length IPRM2 to output first (always an output regardless of ICODE).
	11	Not used.
	10	X Device dependent; usually honest mode.
	9	A Device dependent.
	8	K Device dependent on terminal echo characters.
	7	V Device dependent on lineprinters; do not use code 1 as carriage control.
	6	M Device dependent; usually binary mode.
	5-0	LU Device LU for the operation.

IBFR Data buffer. (Returned for reads.)

ILEN Data length (+ words, - characters).

IPRM1 Optional; can be disk track (for disk transfers), address of additional control buffer (if Z bit is set), or high word of block (for CS/80 disks).

IPRM2 Optional; can be disk sector (for disk transfers), length of additional control buffer (if Z bit is set), or low word of block (for CS/80 disks).

ICLAS Class number; required with Class I/O only. Class number is allocated and assigned an owner by a call to the CLRQ subroutine. The class number can also be allocated by the EXEC 17, 18, 20 call with ICLAS = 0.

Returns:

Normal I/O

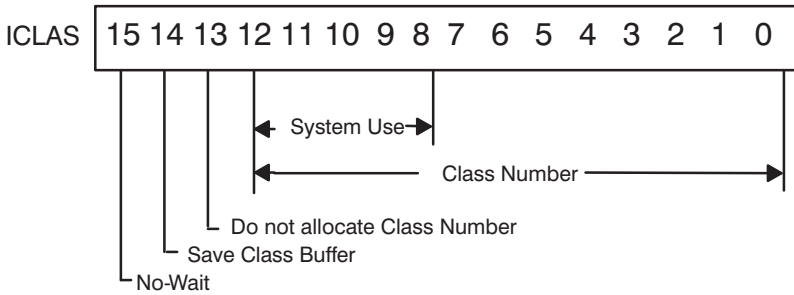
A Status, EQT word 5 (if unbuffered device).
B Transmission log (if unbuffered device).

Class I/O

A 0 – Request completed.
A -1 – No class number (if no wait bit is set).
A 2 – No memory or buffer limit exceeded
(if no wait bit is set).
B Meaningless.

EXEC 21 (I/O, Class Get)

CALL EXEC(21,ICLAS,IBUFR,ILEN[,IP1[,IP2][,IP3])



IBUFR Data buffer.

ILEN Buffer length (+ words, – characters).

IP1 IPRM1 value returned from a class READ/WRITE or CONTROL call.

IP2 IPRM2 value returned from a class READ/WRITE or CONTROL call.

IP3 Returned value of original request code (ICODE).

- 1 17/20 (READ, WRITE/READ)
- 2 18 (WRITE)
- 3 19 (CONTROL)

Returns:

A-Register If data, then A15 = 0 and A = status (EQT word 5).
If no data, and no-wait bit is set, then A15 = 1 and
A = $-(n + 1)$ where n is number of requests made
to class but not yet serviced by driver.

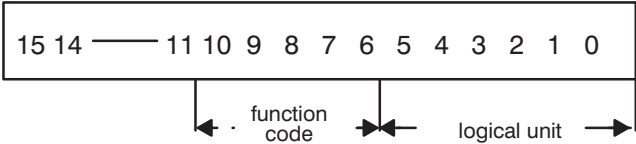
B-Register If data, then B = transmission log (positive words
or characters depending on original request).
If no data, then B is meaningless.

EXEC 3, 19 (I/O Control)

CALL EXEC(ICODE,ICNWD[,IPRAM],ICLAS[,IOP1][,IOP2])

ICODE 3 Control
 19 Class Control

ICNWD Control word; see function codes below for octal bits 6 through 10.



IPRAM Optional or required for some control functions.

TTY: n Space n lines.
 0 No linefeed.

Lineprinter:
 $+n$ Space n lines.
 $-n$ Top-of-form.
 0 No linefeed.

ICLAS Class number; required with class control only. Class number is allocated and assigned an owner by a call to the CLRQ subroutine. The class number can also be allocated by the EXEC 19 call with ICLAS = 0.

IOP1 IOP2
 (When ICODE = 19). Passed through to Class I/O GET request.

Returns:

Normal I/O
 A Status, EQT word 5 (if unbuffered device).
 B Meaningless.

Class I/O
 A Zero.
 B Meaningless.

Function Code:

ICNWD octal bits 6 through 10. See particular driver manual for more information.

- 00 Clear device.
- 01 Write end-of-file (MT,CTU).
- 02 Backspace one record (MT,CTU).
- 03 Forward space one record (MT,CTU).
- 04 Rewind (MT,CTU).
- 05 Rewind standby (MT,REWIND CTU).
- 06 Actual status of device (MT,CTU).
- 07 Set end-of-paper tape.
- 10 Generate paper tape leader.
- 11 List output line spacing, use IPRAM.
- 12 Write gap in case of error (MT).
- 13 Forward space one file (MT,CTU).
- 14 Backward space one file (MT,CTU).
- 15 Conditional top-of-form (LP). Set density (streaming tape).
- 16 Set start/stop mode (streaming tape).
- 17 Set streaming mode (streaming tape).
- 20 Enable terminal (CRT).
- 21 Disable terminal (CRT).
- 22 Set timeout, use IPRAM (CRT).
- 23 Ignore further requests until:
 - a) Device queue is empty.
 - b) Input request encountered.
 - c) Restore control request received.
- 24 Restore output processing.
- 26 Write end-of-data (CTU).
- 27 Locate file number, use IPRAM (CTU).
- 30 Block addressing cached access (CS/80) disks.
- 34 Block addressing (CS/80) disks.

EXEC 6 (Program Completion)

CALL EXEC(6[,INAME][,INUMB][,IPRM1,...,IPRM5])

CALL RMPAR(IPRM1,...IPRM5) (parameter pick-up)

INAME Terminate INAME or if 0, terminate calling program.

INUMB 0 Normal completion (default).

-1 Serially-reusable completion.

1 Terminate saving resources.

2 Terminate on next schedule; save tracks.

3 Terminate immediately and release tracks.

IPRM1-IPRM5

Up to 5 optional parameters passed to caller next time it executes (INAME = 0 only).

Returns:

A-Register Unchanged.

B-Register Unchanged or address of optional parameters (if specified).

EXEC 7 (Program Suspend)

CALL EXEC(7)

If program is rescheduled with a GO command that includes parameters, use RMPAR for parameter pick up.

A-Register Unchanged.

B-Register Unchanged or parameter address.

EXEC 22 (Program Swap Control)

CALL EXEC(22,IOPTN)

IOPTN 0 Swap.

1 Do not swap.

Returns:

A-Register Meaningless.

B-Register Unchanged.

EXEC 8, 9, 10, 23, 24 (Program Schedule)

CALL EXEC(ICODE,INAME[,IPRM1,....,IPRM5] [,IBUFR,ILEN])

ICODE	8	Segment load.
	9	Immediate, wait.
	10	Immediate, no wait.
	23	Queue, wait.
	24	Queue, no wait.

INAME Name of program or segment to be scheduled.

IPRM1–IPRM5

Up to 5 optional parameters passed to program specified in INAME.

IBUFR Buffer to pass to child program. Not used for EXEC 8.

ILEN Length of buffer (+ words, – characters). Child recovers buffer using String Passage (ICODE = 14) EXEC call. Not used for EXEC 8.

Returns:

A-Register	Segment's ID segment address (EXEC 8). 0 if schedule successful or status of child not scheduled (EXEC 9 and 10). Meaningless (EXEC 23 and 24).
B-Register	Unchanged, or address of IPRM1–IPRM5 if they were used.

EXEC 14 (String Passage)

CALL EXEC(14,IRCOD,IBUFR,ILEN)

IRCOD	Retrieve/write code:
	1 Retrieve buffer or command string.
	2 Write buffer to parent.

IBUFR Buffer location.

ILEN Buffer length (+ words, – characters).

Returns:

A-Register	0 = successful; 1 = no string found.
B-Register	Transmission log.

EXEC 13 (Status, Device)

CALL EXEC(13,ICNWD,IST1[,IST2][,IST3])

- ICNWD LU of device.
- IST1 Returned value of EQT word 5, see Device Status Table.
- IST2 Returned value of EQT word 4, see EQT Table.
- IST3 Returned value specifying whether device is “up” or “down”.

Returns: Meaningless.

EXEC 25 (Status, Partition)

CALL EXEC(25,IPART,IPAGE,IPNUM,ISTAT)

- IPART Partition number.
- IPAGE Returned value of starting page number.
- IPNUM Returned value of the number of pages with base page included (-1 returned if illegal partition number).
- ISTAT Return for partition status:



RS RT M S C E ID SEG NO.

- RS 1 if partition reserved.
- RT 1 if partition is real-time.
- M 1 if partition is mother.
- S 1 if partition is subpartition.
- C 1 if chain is in effect.
- E 1 if partition is shareable EMA partition.

Returns:

- A-Register Meaningless.
- B-Register Unchanged.

EXEC 26 (Memory Size)

CALL EXEC(26,IFAW,ILMEM,INPGS[,IMAP])

IFAW	Returned value of first available word address after program.
ILMEM	Returned value, the number of words between end of program and end of program's address space.
INPGS	Returned value, number of pages in partition.
IMAP	Returned value of user map (32-word array).

Returns:

A-Register	Meaningless.
B-Register	Unchanged.

EXEC 11 (Time Request)

CALL EXEC(11,ITIME[,IYEAR])

ITIME	Return for time value as follows: ITIME(1) 10's of milliseconds ITIME(2) Seconds ITIME(3) Minutes ITIME(4) hours ITIME(5) Julian day of year
IYEAR	Returned value of year (for example, 1975) (optional).

Returns:

A-Register	Meaningless.
B-Register	Unchanged.

EXEC 12 (Timed Execution (Absolute Start))

CALL EXEC(12,INAME,IRESL,IMULT,IHRS,IMIN,ISEC,IMSEC)

INAME Schedule INAME or if 0, schedule calling program.

IRESL Resolution code, see initial offset EXEC 12.

IMULT Execution multiple (set = 0 means run once).

IHRS
IMIN
ISEC
IMSEG } Defines absolute start time.

Returns:

A-Register Meaningless.

B-Register Unchanged.

EXEC 12 (Timed Execution (Initial Offset))

CALL EXEC(12,INAME,IRESL,IMULT,IOFST)

INAME Schedule INAME or if 0, schedule calling program.

IRESL Resolution code.

- 1 10's of milliseconds
- 2 Seconds
- 3 Minutes
- 4 Hours

IMULT Execution multiple (0 = run once).

IOFST Relative start time (negative value) from current time.

Returns:

A-Register Meaningless.

B-Register Unchanged.

EXEC 4, 15 (Track Allocation)

CALL EXEC(ICODE,ITRAK,ISTRK,IDISK,ISECT)

ICODE 4 Local
 15 Global

ITRAK Number of tracks:

B15 = 1 Program not suspended if tracks not available.

B15 = 0 Program suspended if tracks not available.

ISTRK Returned value of starting track number (-1 if tracks not available).

IDISK Returned value of disk LU, where tracks were allocated.

ISECT Returned value of number of sectors per track.

Returns: Meaningless.

EXEC 5, 16 (Track Release)

CALL EXEC(ICODE,ITRAK[,ISTRK][,IDISK])

ICODE 5 Local
 16 Global

ITRAK Number of tracks (if ICODE = 5, then -1 = all tracks, ISTRK and IDISK unnecessary).

ISTRK Starting track number.

IDISK Disk LU.

Returns (Local):

A-Register Meaningless.

B-Register Meaningless.

Returns (Global):

A-Register Status:

0 Tracks released.

-1 No tracks released, one in use.

-2 No tracks released, one not global.

B-Register Meaningless.

CLRQ (Class Ownership Management)

CALL CLRQ(IFUNC,ICLAS[,IOP1])

IFUNC	Class management control function.
1	Class ownership assigned.
2	Flush class requests deallocating the class.
3	Flush class requests on LU specified in IOP1.
Bit 14	No-abort bit.
Bit 15	No-wait bit.
ICLAS	Class number.
IOP1	Call dependent parameter; used to describe a program name or LU.

LURQ (Logical Unit Lock Program Call)

CALL LURQ(IOPTN,LUARY,NOLU)

IOPTN	Octal control word as follows:
0xy000	Unlock specified LUs.
1xy000	Unlock all LUs program currently has locked.
0xy001	Lock with wait specified LUs.
1xy001	Lock without wait specified LUs.
x(bit 14)	is no-abort bit; 1 = do not abort.
y(bit 11)	is disk also bit; must be set when specifying a disk LU lock request.
LUARY	Array of LUs to be locked/unlocked. Ignored when IOPTN = 1x0000.
NOLU	Number of LUs to be locked/unlocked. Ignored when IOPTN = 1x0000.

Returns:

A-Register	0	Lock successful.
	-1	RN not available.
	1	LU already locked.

B-Register Unchanged.

REIO, XREIO (Reentrant I/O)

CALL REIO(ICODE,ICNWD,IBUFF,ILEN)

ICODE	1 Read 2 Write
ICNWD	Control word. Specifies the LU (must be non-disk). Can also specify driver-dependent information.
IBUFF	Data buffer.
ILEN	Data length (positive words or negative characters).

Returns:

Same as for EXEC 1 and 2 calls.

XREIO allows access of up to 255 LUs. All other parameters and functions are identical to REIO. Format of ICNWD is same as XLUEX.

RNRQ (Resource Management)

CALL RNRQ(ICODE,IRN,ISTAT)

ICODE	Control word as follows:
Bits	15 No Wait
	14 No Abort
	13 } Not Used
	: }
	: }
	5 Clear } Allocate Option
	4 Global }
	3 Local }
	2 Clear } Set Option
	1 Global }
	0 Local }
IRN	Resource number.
ISTAT	Status word.
	0 Normal deallocate return.
	1 RN is clear (unlocked).
	2 RN is locked locally to caller.
	3 RN is locked globally.
	4 No RN available now.
	6 RN locked locally to other program.
	7 RN was locked globally when request was made.

Returns:

A-Register	Meaningless.
B-Register	Unchanged.

XLUEX (Extended LU EXEC Call)

CALL XLUEX(ICODE,ICNWD,IBFR,ILEN,PRAMS)

Parameters are the same as for EXEC, except ICNWD, which has the format:

Word 1:		Word 2:	
Bit 15	OV bit	Bits 15-6	Same as ICNWD in EXEC call.
Bit 14	Reserved	Bits 5-0	Reserved
Bit 13	0		
Bits 12-8	Reserved		
Bits 7-0	LU		

CI File Handling

Parameters Used in this Section	9-1
Standard FMP Subroutines	9-1
FmpClose	9-1
FmpDcbPurge	9-1
FmpOpen	9-2
FmpOpenScratch	9-3
FmpOpenTemp	9-3
FmpPosition	9-3
FmpPost	9-3
FmpPurge	9-4
FmpRawMove	9-4
FmpRead	9-4
FmpRecordCount	9-5
FmpRecordLen	9-5
FmpRename	9-5
FmpRewind	9-5
FmpSetEof	9-5
FmpSetPosition	9-6
FmpSize	9-6
FmpTruncate	9-6
FmpWrite	9-6
CI File System FMP Subroutines	9-7
Calc_Dest_Name	9-7
FattenMask	9-7
FmpAccessTime	9-7
FmpBuildHierarch	9-7
FmpBuildName	9-8
FmpBuildPath	9-8
FmpCreateDir	9-9
FmpCreateTime	9-9
FmpEndMask	9-9
FmpHierarchName	9-9
FmpInfo	9-9
FmpInitMask	9-10
FmpLastFileName	9-10
FmpLu	9-10
FmpMaskName	9-11
FmpNextMask	9-11

FmpOpenFiles	9-11
FmpOwner	9-12
FmpPagedWrite	9-12
FmpParseName	9-12
FmpParsePath	9-13
FmpProtection	9-13
FmpSetDirInfo	9-13
FmpSetOwner	9-14
FmpSetProtection	9-14
FmpSetWorkingDir	9-14
FmpShortName	9-14
FmpStandardName	9-14
FmpUdspEntry	9-15
FmpUdspInfo	9-15
FmpUnPurge	9-15
FmpUpdateTime	9-16
FmpWorkingDir	9-16
MaskDiscLu	9-16
MaskIsDS	9-16
MaskMatchLevel	9-17
MaskOldFile	9-17
MaskOpenId	9-17
MaskOwnerIds	9-17
MaskSecurity	9-18
WildcardMask	9-18
Utility FMP Subroutines	9-19
DbOpen	9-19
FmpAppend	9-19
FmpBitBucket	9-19
FmpCloneName	9-19
FmpControl	9-20
FmpCopy	9-20
FmpDevice	9-20
FmpDismount	9-21
FmpEof	9-21
FmpError	9-21
FmpExpandSize	9-21
FmpFileName	9-21
FmpInteractive	9-22
FmpIoOptions	9-22
FmpIoStatus	9-22
FmpList	9-23
FmpListX	9-23
FmpMount	9-24
FmpPackSize	9-24
FmpPagedDevWrite	9-25
FmpPaginator	9-26
FmpReadString	9-27
FmpReportError	9-27
FmpRpProgram	9-27
FmpRunProgram	9-28

FmpRwBits	9-28
FmpSetDcbInfo	9-28
FmpSetIoOptions	9-29
FmpSetWord	9-29
FmpUniqueName	9-29
FmpWriteString	9-29
Special-Purpose DS Communication Subroutines	9-30
DsCloseCon	9-30
DsDcbWord	9-30
DsDiscInfo	9-30
DsDiscRead	9-30
DsFstat	9-31
DsNodeNumber	9-31
DsOpenCon	9-31
DsSetDcbWord	9-31

Parameters Used in this Section

The following parameters and conventions are used throughout this section. These parameters and the common ones described in Chapter 1 are not described beneath the calls that use them unless additional information must be given for the call.

- error Returns negative code, or 0 if no error occurs.
- dcb Integer array (at least 16 words) containing file Data Control Block (DCB).
- (*4) Double integer, 4 bytes. If omitted, single integer, 2 bytes (*2), is assumed.

All routines callable as a function are single integer function.

Standard FMP Subroutines

FmpClose

Purpose: Close file (make it inaccessible).

Syntax: error = FmpClose(dcb,error)

FmpDcbPurge

Purpose: Purge a file associated with an open DCB.

Syntax: error = FmpDcbPurge(dcb)

FmpOpen

Purpose: Open file for access. Nonexistent file is created if file descriptor is complete.

Syntax: type = FmpOpen(dcb,error,filedescriptor,options, buffers)

type Nonnegative integer. Return file type or error code.

options Character string. Select file open options from the following:

Access mode:

R Open for reading.

W Open for writing.

File existence:

O Open an existing file.

C Create a new file.

Miscellaneous:

S Open a shared file.

U Open in update mode.

T File is temporary.

E Force LU locking of interactive devices.

F Force type to 1 for unbuffered access.

I Inhibit LU locking of non-interactive devices

X Access extents in type 1 or 2 file.

D file descriptor specifies a directory.

Q Open file quickly, do not record access time.

N File does not contain carriage control.

Options can be in any order, uppercase or lowercase.

buffers DCB buffer size, in range 1 to 127.

FmpOpenScratch

- Purpose:* An interface to FmpOpen; used to standardize scratch file creation.
- Syntax:* type = FmpOpenScratch (dcb,error,filedescriptor,
options,buffers,nameused)
- type See FmpOpen.
- options Same as FmpOpen, plus the following:
Z pass filedescriptor to FmpUniqueName to create a unique scratch file descriptor.
- buffers DCB buffer size, in range 1 to 127.
- nameused Character string. Return the file descriptor used in call to FmpOpen.

FmpOpenTemp

- Purpose:* Open a temporary file.
- Syntax:* type = FmpOpenTemp(dcb,error,filedescriptor,
options,buffers)
- type Same as FmpOpen.
- options Same as FmpOpen except T option is always set automatically.
- buffers Same as FmpOpen.

FmpPosition

- Purpose:* Return current file position.
- Syntax:* error = FmpPosition(dcb,error,record,position)
- record(*4) Current record number.
- position(*4) Current internal file position.

FmpPost

- Purpose:* Post data to a file.
- Syntax:* error = FmpPost(dcb,error)

FmpPurge

Purpose: Purge a file.

Syntax: error = FmpPurge(filedescriptor)

FmpRawMove

Purpose: Read or write data to a disk file starting at a specified internal file position.

Syntax: length = FmpRawMove(dcb,error,positn,buffer,
maxlength,how)

length An integer that returns the number of words successfully transferred to or from the disk file.

positn(*4) The desired internal file position.

buffer A word-aligned integer buffer that either contains the data to be transferred (*how*=2), or returns the data being transferred (*how*=1).

maxlength An integer that contains the number of words to be transferred.

how An integer that specifies the direction of the transfer.

1 Read data from the file into 'buffer'.

2 Write data from 'buffer' to the file.

The internal file position after the call is undefined. It is the caller's responsibility to reset the internal file position after the call.

FmpRead

Purpose: Read from a file.

Syntax: length = FmpRead(dcb,error,buffer,maxlength)

length Number of bytes actually read, or negative error code. If more than 32767 bytes are read, length may be negative although no error occurred.

buffer Word-aligned buffer into which data file is to be transferred.

maxlength Maximum number of bytes to read. Treated as unsigned single integer from 0 to 65534.

FmpRecordCount

Purpose: Return number of records in file.

Syntax: error = FmpRecordCount(filedescriptor,nrecords)

nrecords(*4) Number of records in file.
For file types 1 and 2, this is the maximum number of records that the file can accommodate.
For file types 3 and above, this is the number of records before EOF (may be inaccurate if file is open for writing).

FmpRecordLen

Purpose: Return length of longest record in file.

Syntax: error = FmpRecordLen(filedescriptor,len)

len Length of longest record in file. For file types 3 and above, length of longest record ever written to file is returned even if it has been overwritten.

FmpRename

Purpose: Change file name.

Syntax: error = FmpRename(name1,err1,name2,err2)

name1 Name of existing, closed file.
err1 Error associated with the file name1.
name2 New name, including security code and directory.
err2 Error associated with the file name2.

FmpRewind

Purpose: Position file at its first record.

Syntax: error = FmpRewind(dcb,error)

FmpSetEof

Purpose: Set end-of-file mark at current file position.

Syntax: error = FmpSetEof(dcb,error)

FmpSetPosition

Purpose: Change file position.

Syntax: error = FmpSetPosition(dcb,error,record,position)

record(*4) Number of record at which file is to be positioned.

position(*4) If positive, desired internal file position; if negative, desired record number.

FmpSize

Purpose: Return physical file size.

Syntax: error = FmpSize(filedescriptor,size)

size(*4) File size in blocks.

FmpTruncate

Purpose: Truncate file.

Syntax: error = FmpTruncate(dcb,error,blocks)

blocks(*4) Minimum number of blocks to which file will be truncated.

FmpWrite

Purpose: Write to a file.

Syntax: length = FmpWrite(dcb,error,buffer,maxlength)

length Number of bytes actually transferred, or a negative error code. If more than 32767 bytes are transferred, length may be negative although no error occurred.

buffer Word-aligned buffer containing data to be transferred.

maxlength Maximum number of bytes to write; interpreted as unsigned one-word integer from 0 to 65534.

CI File System FMP Subroutines

Calc_Dest_Name

Purpose: Create destination file name from a file name, match level, and destination mask.

Syntax: Call Calc_Dest_Name(sourcename,matchlevel,destmask,destname)

sourcename	Character string containing full source file descriptor.
matchlevel	Output of MaskMatchLevel routine.
destmask	Character string specifying destination mask.
destname	Character string that returns full destination file descriptor.

FattenMask

Purpose: Modify mask.

Syntax: Call FattenMask(mask,how)

mask	Character string specifying mask to be modified.
how	Integer specifying how to modify mask. If bit 0 is set, “D” is appended to the qualifier; if bit 1 is set and the mask is blank, neither the name nor the type extension will have “@” inserted.

FmpAccessTime

Purpose: Return time of last access for a file.

Syntax: error = FmpAccessTime(filedescriptor,time)

time(*4)	Return time of last access, expressed as number of seconds since January 1, 1970.
----------	---

FmpBuildHierarch

Purpose: Build file descriptor in CI format.

Syntax: Call FmpBuildHierarch(filedescriptor,dirpath,name,typex,qual,sc,type,size,rl,ds)

Parameters are the same as for FmpBuildPath.

FmpBuildName

Purpose: Build file descriptor from given file specifiers.

Syntax: Call `FmpBuildName(filedescriptor,name,typex,sc,dir,type,size,rl,ds)`

filedescriptor	63-character string containing returned file descriptor.
name	Character string (up to 63 characters) specifying subdirectories (if any) and the file name.
typex	Character string (up to 4 characters) specifying the file type extension.
sc	Security code (FMGR files).
dir	Character string (up to 16 characters) specifying the global directory name.
type	FMP file type.
size	File size in blocks.
rl	Record length.
ds	Character string (up to 63 characters) specifying DS node name, a user name, or both.

FmpBuildPath

Purpose: Build character string file mask or file descriptor from given file specifiers.

Syntax: Call `FmpBuildPath(filedescriptor,dirpath,name,typex,qual,sc,type,size,rl,ds)`

File specifiers are as described for `FmpBuildName`, except:

dirpath	Character string (up to 63 characters) naming directory/subdirectory path. Dirpath must end with “/”, and must have “/” between each of the directories and subdirectories.
name	Character string specifying a file name (up to 16 characters).
qual	Character string mask qualifier (up to 40 characters).

FmpCreateDir

Purpose: Create directory.

Syntax: error = FmpCreateDir(name,lu)

name Name of directory to be created.

lu Disk LU on which to create directory.

FmpCreateTime

Purpose: Return time when a file was created.

Syntax: error = FmpCreateTime(filedescriptor,time)

time(*4) Returned time when file was created, expressed in seconds since January 1, 1970.

FmpEndMask

Purpose: Close files associated with mask search.

Syntax: Call FmpEndMask(dirdcb)

dirdcb Integer array initialized by FmpInitMask.

FmpHierarchName

Purpose: Convert file descriptor to CI file format.

Syntax: Call FmpHierarchName(filedescriptor)

FmpInfo

Purpose: Return directory information for a file.

Syntax: error = FmpInfo(dcb,error,info,flag)

info 32-word integer array in which directory information is returned.

flag 0 to FMGR file, nonzero for CI file.

FmpInitMask

Purpose: Initialize file structures to FMP mask calls.

Syntax: error = FmpInitMask(dirdcb,error,mask,
diropenname,dcblen[,msc])

dir dcb Control array, to be used only with FmpNextMask.
Must be at least 372 words.

mask Character string specifying set of files. Format is:
dirpath/name.typex,qual:sc:dir:type:size:rl

diropenname Returns character string directory path.

dcb len Length of dir dcb, in words.

msc Master security code, allows security code on FMGR
files to be viewed.

FmpLastFileName

Purpose: Return last file name in path.

Syntax: subroutine FmpLastFileName (filedescriptor,lastname)

lastname Returns filename, a portion of file descriptor.

FmpLu

Purpose: Return LU number of file or device associated with
specified DCB.

Syntax: lu = FmpLu(dcb)

dcb Integer array containing the DCB for the file.

lu Integer indicating the LU number of the file or device
associated with the specified DCB.

If the DCB is associated with a type zero file, the value returned in the lu parameter is the device LU number. If the DCB is associated with a disk file, the value returned is the LU of the disk on which the file resides. If the specified DCB is not open, a -11 (DCB not open error) is returned.

FmpMaskName

- Purpose:* Build full name for file that matches mask.
- Syntax:* Call `FmpMaskName(dirdcb,newname,entry,curpath)`
- `dirdcb` Control array initialized by `FmpInitMask`.
- `newname` Character string that returns the file descriptor.
- `entry` 32-word directory entry from `FmpNextMask`.
- `curpath` Character string directory path from `FmpNextMask`.

FmpNextMask

- Purpose:* Return directory entry of next file that matches mask.
- Syntax:* `more = FmpNextMask(dirdcb,error,curpath,entry)`
- `more` Boolean variable indicating whether search can continue. True if there is another entry to be searched, whether or not an error occurred (if error did occur, current entry is invalid). False if error prevents continuation of search, or when search is complete.
- `dirdcb` Control array initialized by `FmpInitMask`.
- `curpath` Character string directory path returned.
- `entry` 32-word array that returns directory entry for each file found.

FmpOpenFiles

- Purpose:* Indicate open files in a directory.
- Syntax:* `error = FmpOpenFiles(dcb,error,loc,flag)`
- `dcb` Directory open for reading.
- `loc` Directory position of next open file. Caller initializes it to 0 to indicate that this is the first call. When all open files in directory are reported, `loc = -1`.
- `flag` Return flag value for a file.

FmpOwner

Purpose: Return name of directory or volume owner.

Syntax: error = FmpOwner(dir,owner)

dir Directory name or volume number.

owner Character string returned showing the logon name of the user who owns directory or volume.

FmpPagedWrite

Purpose: Write data to a file using FmpPaginator to break output into screen pages for terminal devices.

Syntax: status = FmpPagedWrite (dcb, error, buffer, length, pageinfo)

buffer Character string specifying word-aligned buffer containing data to be transferred.

length Integer, from 0 to 65534, specifying the number of bytes to write. For a value larger than 32767, set length to desired number of bytes minus 65534.

pageinfo Five-word array holding paging information for FmpPaginator; if first word is zero, the default values are used; see FmpPaginator routine description in Utility Subroutines section.

status Integer returning one of the following:

0 ready for next line.

1 abort listing.

or, negative FMP error code.

FmpParseName

Purpose: Separate character string file descriptor into file specifiers.

Syntax: Call FmpParseName(filedescriptor,name,typex,sc, dir,type,size,rl,ds)

Parameters are the same as for FmpBuildName.

FmpParsePath

Purpose: Separate character string file mask or file descriptor into file specifiers.

Syntax: Call FmpParsePath(filedescriptor,dirpath,name,typex,qual,sc,type,size,rl,ds)

Parameters are the same as for FmpBuildPath.

FmpProtection

Purpose: Return access available for file or directory (R = read, W = write, RW = both).

Syntax: error = FmpProtection(filedescriptor,owneraccess, othersaccess[,groupaccess])

owneraccess Owner's file access rights.

othersaccess File access rights of users other than owner.

groupaccess File access rights of members of owner's group.

FmpSetDirInfo

Purpose: Change directory information.

Syntax: error = FmpSetDirInfo(dcb,error,ctime,atime, utime,bbit,prot[,option])

ctime(*4) Create time.

atime(*4) Access time.

utime(*4) Update time.

bbit Backup bit.

prot File protection, where:

Bit 0 = 1 general user may write

Bit 1 = 1 general user may read

Bit 2 = 1 owner may write

Bit 3 = 1 owner may read

Bit 6 = group may write only if G specified

Bit 7 = group may read only if G specified

Any bit set to zero denies permission.

option optional string that determines interpretation of *prot*.

G = *prot* contains valid group bits. If G is not specified, or if the parameter is not present, the group bits of *prot* are ignored; general user bits (bits 0 and 1) are used for group bits.

If a parameter is negative, the corresponding value in the directory entry is not changed.

FmpSetOwner

Purpose: Change name of directory or volume owner. Caller must own directory or be a superuser.

Syntax: error = FmpSetOwner(dir,err1,owner,err2)

dir Directory name or volume number.
err1 Return errors associated with dir.
owner Name of new owner.
err2 Return errors associated with owner.

FmpSetProtection

Purpose: Change access available for file or directory (R = read, W = write, RW = both).

Syntax: error = FmpSetProtection(filedescriptor,
owneraccess,othersaccess[,groupaccess])

owneraccess Owner's file access rights.
othersaccess File access rights of users other than owner.
groupaccess File access rights of members of owner's group.

FmpSetWorkingDir

Purpose: Change working directory.

Syntax: error = FmpSetWorkingDir(filedescriptor)

FmpShortName

Purpose: Return short version of file descriptor, without type, size, or record length.

Syntax: error = FmpShortName(dcb,error,filedescriptor)

FmpStandardName

Purpose: Convert file descriptor to standard format (that is, subdir/filename::dir).

Syntax: Call FmpStandardName(filedescriptor)

FmpUdspEntry

Purpose: Return the directory name for the specified entry and User-Definable Directory Search Path (UDSP).

Syntax: error = FmpUdspEntry(udspnum,entnum,dirname,
error)

udspnum Integer specifying the UDSP number.

entnum Integer specifying the entry for the UDSP number.

dirname Character string returning the directory name for the specified entry in the specified UDSP.

error Integer returning one of the following values:

- 0 No error occurred
- 1 Not under session control
- 2 UDSP tables not set up correctly

If the entry is undefined, or if udspnum and entnum are out of bounds with the definition for the session, error returns -247.

FmpUdspInfo

Purpose: Returns the current User-Definable Directory Search Path (UDSP) information for your session.

Syntax: error = FmpUdspInfo(udsp,depth,next,error)

udsp Integer returning the number of UDSPs defined for the current session.

depth Integer returning the UDSP depth defined for the current session.

next Integer returning the next available UDSP. Next is set to zero if all UDSPs are defined.

error Integer returning one of the following values:

- 0 No error occurred
- 1 Not under session control
- 2 UDSP tables not set up correctly

FmpUnPurge

Purpose: Restore directory information for a file (undo purge).

Syntax: error = FmpUnPurge(filedescriptor)

FmpUpdateTime

Purpose: Return last update time for a file.

Syntax: error = FmpUpdateTime(filedescriptor,time)

time(*4) Return time of last update, expressed in seconds since January 1, 1970.

FmpWorkingDir

Purpose: Return current working directory.

Syntax: error = FmpWorkingDir(name[,format])

name Return name of current working directory.

format Defines format of the directory string being returned:

- 0 (default) if a working directory is a global directory, it is returned in the trailing directory format (“:dir”); otherwise, returned in hierarchical format with no trailing slash.
- 1 working directory returned in hierarchical format with no trailing slash.
- 2 working directory returned in hierarchical format with a trailing slash.

MaskDiscLu

Purpose: Return the disk LU of the last file returned by FmpNextMask

Syntax: diskLu = MaskDiscLu(dirdcb)

diskLu Integer returning the disk LU.

dirdcb Integer array, initialized by FmpInitMask.

MaskIsDS

Purpose: Determines if masking is searching a remote file system.

Syntax: bool = MaskIsDS(dirdcb[,dsinfo])

bool Boolean variable that returns TRUE (negative value) if masking is searching a remote file system; otherwise, it returns FALSE (non-negative value).

dirdcb Control array, initialized by FmpInitMask.

dsinfo Character string that returns the DS information of the mask.

MaskMatchLevel

Purpose: Return directory level of last file matched.

Syntax: matchlevel = MaskMatchLevel(dirdcb)

matchLevel Directory level in which last file was matched.

dirdcb Integer array initialized by FmpInitMask.

MaskOldFile

Purpose: Determine if file is a FMGR file.

Syntax: bool = MaskOldFile(dirdcb)

bool True if last file returned by FmpNextMask is a FMGR file.

dirdcb Integer array initialized by FmpInitMask.

MaskOpenId

Purpose: Return D.RTR open flag of last file returned by FmpNextMask.

Syntax: opendir = MaskOpenId(dirdcb)

opendir D.RTR open flag of last file returned by FmpNextMask; 0 if file is closed.

dirdcb Integer array initialized by FmpInitMask.

MaskOwnerIds

Purpose: Return the owner and group IDs for the last file returned by FmpInitMask.

Syntax: call MaskOwnerIDs(dirdcb,ownerid,groupid)

dirdcb Integer array; initialized by FmpInitMask.

ownerid Integer returning ID number of the file owner.

groupid Integer returning ID number of the associated group of the file.

MaskSecurity

- Purpose:* Return security code of last FMGR file returned by FmpNextMask.
- Syntax:* `seccode = MaskSecurity(dirdcb)`
- `seccode` Security code of last file returned by FmpNextMask if file was an FMGR file; 0 for FMP file.
- `dirdcb` Integer array initialized by FmpInitMask.

WildCardMask

- Purpose:* Check for wildcard characters in mask.
- Syntax:* `wild = WildCardMask(mask)`
- `wild` True if mask refers to more than one file; false otherwise.
- `mask` Character string containing mask to be checked.

Utility FMP Subroutines

DcbOpen

Purpose: Indicate whether DCB is open.

Syntax: error = DcbOpen(dcb,error)

error 0 if DCB is open; negative error code if not.

FmpAppend

Purpose: Position file at EOF mark.

Syntax: error = FmpAppend(dcb,error)

FmpBitBucket

Purpose: Determine if the type 0 file associated with the specified DCB is LU 0 (the bit bucket).

Syntax: bool = FmpBitBucket(dcb)

dcb Integer array containing the DCB for the type 0 file.

bool Boolean. A flag that is set to TRUE (negative value) if the DCB is open and associated with a type zero file, and the device is LU 0; otherwise, bool is set to FALSE (non-negative value).

FmpCloneName

Purpose: Generate program clone names that can be used by FmpRpProgram.

Syntax: Call FmpCloneName(name,init)

name Character string specifying the program name to be cloned. The specified name is modified by the system and returned to the calling program.

init Logical indicating if the current call is the first call to FmpCloneName.

FmpDismount

Purpose: Dismount a volume.

Syntax: error = FmpDismount(lu)

lu LU of volume to be dismounted.

FmpEof

Purpose: Return position of EOF mark.

Syntax: error = FmpEof(filedescriptor,eofpos)

eofpos(*4) Current internal file position.

FmpError

Purpose: Return error message for FMP error code.

Syntax: Call FmpError(error,message)

message Character string that returns an error message. If no message is associated with the error identified by the error parameter, a generic error message in the form “FMP error -xxx” is returned.

FmpExpandSize

Purpose: Unpack file size word into double integer.

Syntax: blocks = FmpExpandSize(size)

blocks(*4) Number of blocks in file, in double integer.

size File size, in one word.

FmpFileName

Purpose: Return full path name of file.

Syntax: error = FmpFileName(dcb,error,filedescriptor)

FmpInteractive

Purpose: Indicate if DCB is associated with an interactive device.

Syntax: bool = FmpInteractive(dcb)

bool True (-1) if DCB is associated with an interactive device; false (0) otherwise.

FmploOptions

Purpose: Return I/O option word.

Syntax: error = FmploOptions(dcb,error,options)

options Character string that selects copy options, which are:

A ASCII.
B Binary.
D Overwrite existing file.
P Purge source after copy.

FmploStatus

Purpose: Return A- and B-Register values.

Syntax: Call FmploStatus(areg,breg)

areg Value of A-Register.

breg Value of B-Register.

FmpList

Purpose: List file to specified LU.

Syntax: error = FmpList(filedescriptor,lu,option,rec1,rec2)

lu Output LU.

option Character string that selects output format:

- A ASCII.
- B Binary output displayed as octal.
- C FORTRAN carriage control chars in column 1.
- M Count lines longer than 80 characters as multiple lines for page breaking.
- Q Quiet; do not record access time of file.
- T Truncate trailing blanks on lines.

Output format defaults are:

File Type:	Format:
0, 3, 4	ASCII
1, 2, 5 and up	Binary

rec1(*4) First record to be listed.

rec2(*4) Last record to be listed.

If rec1 = 0 and rec2 = 0, whole file is listed.

FmpListX

Purpose: Extended version of FmpList. Allows the caller to provide a buffer so that lines longer than 256 bytes can be listed. Also allows listing to be sent to a file.

Syntax: error = FmpListX(sourcefile,destfile,options,startrec, endrec,buffer,maxlength)

sourcefile Name of the file to be listed.

destfile Name of the destination listing file.

options Character string that selects the output format and options. The values are as follows:

- A ASCII output.
- B Binary output displayed as octal.
- C FORTRAN carriage control chars in column 1.
- M Count lines longer than 80 characters as multiple line for page breaking.
- Q Quiet – do not record access time of file.
- T Truncate trailing blanks on line.

File types 0, 3, and 4 default to A; all other file types default to B.

startrec	First record number to be listed.
endrec	Last record number to be listed.
buffer	Buffer for transporting records between sourcefile and destfile.
maxlength	Maximum number of bytes that may be contained in buffer.

FmpMount

<i>Purpose:</i>	Mount a volume.
<i>Syntax:</i>	error = FmpMount(lu,flag,blks)
lu	LU of disk volume.
flag	Determine whether to initialize disk before mounting it. The values of flag are:
	0 Do not initialize before mounting.
	1 Initialize if disk does not have valid a directory.
	2 Initialize disk before mounting.
blks	Number of blocks to leave free at beginning of volume.

FmpPackSize

<i>Purpose:</i>	Pack double integer file size into one word.
<i>Syntax:</i>	size = FmpPackSize(doublesize)
size	Return file size in one word.
doublesize(*4)	File size in double integer.

FmpPagedDevWrite

Purpose: Perform XLUEX(2) write to a device with page breaks for interactive devices.

Syntax: status = FmpPagedDevWrite(cntwd, buffer, length,
pageinfo)

cntwd Two-word integer XLUEX control word specifying LU from 0 to 255 to be written to.

buffer Integer array containing data to be transferred.

length Positive integer indicating number of words, negative integer indicating number of bytes.

pageinfo Five-word array holding paging information for FmpPaginator; see FmpPaginator routine description.

status Integer returning:
0 ready for next line.
1 abort listing.

FmpPaginator

Purpose: Prompts for page breaks when FmpList, FmpPagedWrite, and FmpPagedDevWrite routines are used.

Syntax: status = FmpPaginator(lu, paginfo)

lu Integer containing LU number, from 0 to 255, to be prompted.

paginfo Five-word array holding the following information:

word usage

- 0 page size in lines, or, 0 for default values.
- 1 lines to print before page break, or -1 for no paging.
- 2 address of prompt buffer.
- 3 length of prompt buffer in bytes.
- 4 flags:
 - bit meaning (if set)**
 - 15 do not print current page
 - 0 use unbuffered I/O

If word 0 is zero, the following default values are given to each word:

word default value

- 0 22 lines per page
- 1 22 lines listed before breaking
- 2 address of string, "More..."
- 3 7 characters (length of string)
- 4 zero (no special flags set)

status Returns one of the following values:

- 0 ready for next line.
- 1 abort listing.
- 2 continue listing, but skip this line.

Descriptions of valid responses to prompts are as follows (responses may be preceded by <n>, a number from 1 to 32767):

character	action
<space>	list another page or <n> lines.
<return>	list remaining text without paging.
A or Q	abort listing and return 1.
+	list next line or skip <n> lines and list one more line.
P	set page size to <n> lines and list another page.
Z	suspend calling program (restart with GO command).

FmpReadString

Purpose: Read character string from file.

Syntax: length = FmpReadString(dcb,error,string)

length Return positive number of bytes transferred, or negative error code.

string Character string (up to 256 bytes) into which data is transferred.

FmpReportError

Purpose: Print error message for FMP error code on LU 1.

Syntax: Call FmpReportError(error,filedescriptor)

filedescriptor File descriptor to be included with error message.

FmpRpProgram

Purpose: Restore program

Syntax: error = FmpRpProgram(filedescriptor,rpname,
options,error)

rpname Character string that either specifies or returns program name.

options Character string containing “C”, “P”, or both to select either of the following options:

- C Create clone name if specified or assigned name is already assigned to an RP'd program. Program is not cloned if:
 - System program has the assigned or specified name.
 - Another program has the assigned or specified name, but it is not RP'd.
 - No program with that name is currently RP'd.
- P Do not release ID segment when program completes.

FmpRunProgram

- Purpose:* Schedule a program.
- Syntax:* error = FmpRunProgram(string,parms,runname
[,alterstring])
- string Character string specifying runstring.
- parms Return RMPAR parameters from program when it completes. If string specifies XQ, these parameters are meaningless.
- runname Character string that returns true name used to schedule program.
- alterstring Boolean variable indicating how FMP is to handle string variables. If false, FMP does not alter the string. If true, FMP converts string to blanks with a comma.

FmpRwBits

- Purpose:* Check string for letters R and W.
- Syntax:* value = FmpRwBits(string)
- string Character string (up to 256 bytes).
- value One of the following, depending upon string content:
- 0 Neither present.
 - 1 W, but not R present.
 - 2 R, but not W present.
 - 3 R and W present.

FmpSetDcblInfo

- Purpose:* Change information in DCB
- Syntax:* error = FmpSetDcblInfo(dcb,error,records,eofpos,
recLen)
- records(*4) Number of records in the file, plus one.
- eofpos(*4) Current internal file position.
- recLen Length of longest record, in words.

FmpSetloOptions

Purpose: Change I/O option word.

Syntax: error = FmpSetloOptions(dcb,error,options)

options Same as for FmpIoOptions.

FmpSetWord

Purpose: Change file position.

Syntax: error = FmpSetWord(dcb,error,position,how)

position(*4) Desired file position.

how Specify whether file system should create extent to contain new position if it is outside the existing file area.

1 = extent creation is not permitted;

2 = extent creation is permitted.

FmpUniqueName

Purpose: Create and return unique file name.

Syntax: Call FmpUniqueName(prefix,uniquename)

prefix Prefix for unique file name.

uniquename Return file name that is unique within system that contains no files from another system.

FmpWriteString

Purpose: Write character string to file.

Syntax: length = FmpWriteString(dcb,error,string)

length Return length of record written to file, or negative error code.

string Character string (up to 256 bytes) from which data is transferred.

Special-Purpose DS Communication Subroutines

DsCloseCon

Purpose: Close connection set up to DsOpenCon.
Syntax: error = DsCloseCon(conn)
conn Connection number of disk to be closed.

DsDcbWord

Purpose: Return first word of DCB as it would appear if file associated with it were not opened via DS.
Syntax: error = DsDcbWord(conn,word)
conn Connection number of system.
word Return first word of DCB.

DsDiscInfo

Purpose: Return number of tracks and blocks per track for specified disk volume.
Syntax: error = DsDiscInfo(conn,lu,ntracks,bpert)
conn Connection number of system containing disk.
lu LU of disk volume.
ntracks Return number of tracks on disk volume.
bpert Return number of blocks per track on disk.

DsDiscRead

Purpose: Read disk.
Syntax: error = DsDiscRead(conn,buf,len,track,sector)
conn Connection number of disk. Must have been set by DsSetDcbWord.
buf Buffer to hold data read from disk.
len Number of characters to read (up to 4096).
track Track from which to read.
sector 64-word sector from which to read (even number).

DsFstat

Purpose: Perform FSTAT call for specified system.

Syntax: error = DsFstat(conn,buffer,lu[,format[,iop]])

conn Connection number of system.

buffer At least 256 words.

lu LU of system on which to perform FSTAT.

format Same as for FSTAT.

iop Same as for FSTAT.

DsNodeNumber

Purpose: Returns node number associated with the specified file.

Syntax: node = DsNodeNumber(filedescriptor)

node Returns the number of the node associated with the specified file. Zero is returned if the file is not remote.

filedescriptor 63-character string that specifies the name of a file.

DsOpenCon

Purpose: Open connection to remote user account/node.

Syntax: error = DsOpenCon(string,conn)

string Remote user account name, node name, or both, along with required delimiters. Must not contain a file name, only DS information.

conn Return connection number.

DsSetDcbWord

Purpose: Change first word of DCB so that DsDiscRead works.

Syntax: error = DsSetDcbWord(conn,word)

conn Connection number of disk to be read by DsDiscRead.

word Word to be changed.

FMGR File Handling

Parameters	10-1
APOSN and EAPOS	10-2
CLOSE and ECLOS	10-2
CREAT and ECREA	10-3
CRETS	10-3
FCONT	10-4
FSTAT	10-4
IDCBS	10-5
LOCF and ELOCF	10-6
NAMF	10-6
OPEN and OPENF	10-7
POSNT and EPOSN	10-8
POST	10-8
PURGE	10-9
READF and EREAD	10-9
RWNDF	10-9
WRITF and EWRT	10-10

Parameters

NOTE

The FMP calls beginning with E (e.g., ECREA) can define larger files, up to 32767x128 blocks. The FMP calls not beginning with E (e.g., CREAT) can only define files up to 16383 blocks, and 32767 records.

IDCB	A 144 -word or longer, array used as the Data Control Block (DCB).
IERR	Error return, see the negative FMGR error codes for meaning. If call is successful. OPEN,OPENF IERR = file type. CREAT IERR = number of sectors.
INAM	Six ASCII characters. First character not a blank or number, no embedded blanks, and (+,-:) are not allowed. All six placed must be accounted for, and a FORTRAN DATA statement can be used to specify INAM.
IBUF	User buffer.
ISC	File security code: <0 Read/Write protected. =0 Not protected (default). >0 Write protected only.
ICR	Cartridge reference: >0 Cartridge reference number. <0 Logical Unit number. =0 First one found (default). Order of search; private cartridges, then group cartridges, then system cartridges.
IREC	Next record number, double word for "E" type calls.
IOFF	Block offset of next record.
IRB	Relative block address of next record, double word for "E" type calls.
IDCBZ	Actual size of DCB in words (only when IDCB > 144).
<>	Parameters in angle brackets are optional in some cases and required in others. Refer to call parameter description for details.

APOSN and EAPOS

```
      APOSN
CALL   (IDCB,IERR,IREC[,IRB[,IOFF]])
      EAPOS
```

Position a disk file (typically type 3) to a known record address. Record addresses are usually obtained through LOCF for APOSN, and ELOCF or EAPOS. IRB and IOFF are required for files with variable length records.

CLOSE and ECLOS

```
      CLOSE
CALL   (IDCB<,IERR>[,ITRUN])
      ECLOS
```

Close DCB and make file available to others, can also truncate file size.

- | | |
|-------|---|
| ITRUN | One-word variable for CLOSE, double word variable for ECLOS. |
| | +n Number of blocks to be deleted from the end of the file when it is closed. |
| | -n Retain main file, delete extents. |
| | 0 Standard close (default). |
| IERR | Required when ITRUN is specified. Optional otherwise. |

CREAT and ECREA

```
CALL      CREAT
          (IDCB,IERR,INAM,ISIZE,ITYPE[,ISC][,ICR]
          ECREA                                [,IDCBZ][,JSIZE])
```

Create a disk file.

- IERR Error return. If call is successful, IERR = number of sectors.

- ISIZE Two-entry array describing file size: for CREAT a two-word array, for ECREA a double-word integer for each entry.
 - first entry File size in blocks.
 - second entry Record length in words (used for type 2 files only).

- ITYPE File type (1-32767).

- JSIZE Created file size in sectors; optional double word parameter returned by ECREA only.

CRETS

```
CALL CRETS (IDCB,IERR,NUM,INAM[,ISIZE][,ITYPE][,ISC]
            [,ICR][,IDCBZ][,JSIZE])
```

Create a temporary or scratch disk file by making an entry in the File Directory and allocating disk space for the file. CRETS can define files up to 32767x128 blocks in size.

- NUM Scratch file number, a one-word integer 0-99.

- ISIZE A double-word integer for each entry.
 - first entry File size in blocks.
 - second entry Record length in words (used for type 2 files only).

- ITYPE File type (1-32767).

- JSIZE Created file size in sectors; optional double word parameter returned if call was successful.

FCONT

CALL FCONT(IDCB,IERR,ICON1 [,ICON2])

Control I/O functions on a non-disk type 0 file.

- ICON1 Control word; refer to Chapter 8, I/O Control, EXEC 3 call for function codes.
- ICON2 Additional control or status returned. Required if function code in ICON1 is 11B or 27B.

FSTAT

CALL FSTAT(ISTAT [,ILEN] [,IFORM] [,IOP] [,IADD])

Return status of mounted cartridges.

- ISTAT Cartridge status buffer returned as FORMAT I or FORMAT II.

FORMAT I		
WORD	CONTENTS	CARTRIDGE
1 2 3 4	Logical unit number Last FMP track Cartridge Reference Number Lock Word	First Cartridge
5 6 7 8	Logical unit number Last FMP track Cartridge Reference Number Lock Word	Second Cartridge
9 . . .	Logical unit number
	0 no more disks	

where: Lock word is ID segment address of locking program or 0 (not locked).

FORMAT II		
WORD	CONTENTS	CARTRIDGE
1 2 3 4	Lock Word Logical Unit # Last FMP track Cartridge Reference Number ID	First Cartridge
5 6 7 8	Lock Word Logical Unit # Last FMP track Cartridge Reference Number ID	Second Cartridge
9 . . .	Lock Word Logical Unit #
	0 no more disks	

where: Lock word is the offset of the ID segment in the Keyword Table or 0 (not included). ID identifies who mounted the cartridge.

ILEN Length in words of status buffer (default = 125).

IFORM Zero for FORMAT I. Nonzero for FORMAT II.

IOP Type of cartridges to return information about:

- 1 All cartridges mounted to the system.
- 0 (Under session) all private, group, and system cartridges mounted to that session.
- 0 (Non-session) mounted system and non-session cartridges.

IADD 0 if entire cartridge list was returned. Nonzero if entire cartridge list could not be returned.

IDCBS

ISIZE = IDCBS(IDCBS)

Return actual DCB buffer area used (use only if DCB > 144).

LOCF and ELOCF

```
CALL      LOCF
          (IDCB,IERR,IREC[,IRB][,IOFF][,JSEC][,JLU]
          ELOCF                               JTY][,JREC])
```

Retrieve status and location information from the Data Control Block (DCB) on an open file.

JSEC	File size in sectors; one-word variable for LOCF, double-word variable for ELOCF.
JLU	File LU.
JTY	File type.
JREC	Optional return for:
	record length Type 1 or 2 files.
	read/write code Type 0 files.
	meaningless Type 3 and above.

NAMF

```
CALL NAMF(IDCB,IERR,INAM,MNAM[,ISC][,ICR])
```

Close the DCB, if open, and rename file INAM to MNAM.

(Refer to Parameter section for description of common parameters.)

OPEN and OPENF

CALL OPEN (DCB, IERR, INAM[, IOPTN][, ISC][, ICR][, IDCBZ]
OPENF

Open a file for access.

IERR Error return. If call is successful IERR = file type.
INAM ASCII file name, or an integer containing a binary LU (OPENF only).
IOPTN Open options, specified by octal value representing one-word variable after bits have been appropriately set:

E (bit 0) = 0 File opened exclusively to calling program; locked if type 0.
 = 1 File can be shared by up to seven programs.
U (bit 1) = 0 File opened for standard (not update) write.
 = 1 File opened for update.
T (bit 2) = 0 File opened for standard (not update) write.
 = 1 Force file type to 1.
X (bit 5) = Permit extents on type 1 and 2 files.
Bit F is ignored for all by type 0 files.
F (bit 3) = 0 Use function code defined at creation.
 = 1 Use function code defined in bits 6-10 of IOPTN.

Bits 6-10 correspond exactly to function code used for read or write call (EXEC 1 or 2).

If omitted or 0, file is opened exclusively to the calling program, for standard sequential output, with the file type or function code assigned at creation.

POSNT and EPOSN

```

    POSNT
CALL   (IDCB,IERR,NUR[,IR])
    EPOSN
  
```

Position files relative to current file position or to a specific record number in any file type.

- NUR Record position, a one-word variable for POSNT or double-word variable for EPOSN.
- IR Position mode flag, the relationship between NUR and IR is:

NUR	IR = 0 OR OMITTED RELATIVE POSITION	IR ≠ 0 ABSOLUTE POSITION
NUR > 0	Position forward number of records specified.	Position to record number specified.
NUR = 0	No operation.	No operation
NUR < 0	Position backward number of records specified.	Error

POST

```
CALL POST(IDCB[,IERR])
```

Write contents of DCB to the disk, and save records in a file opened for non-exclusive use. To lock the file for exclusive use with RNRQ call, use the following sequence:

1. Call OPEN.
2. Read file to pick up resource number.
3. Call POST to clear DCB, no data is transferred.
4. Call RNRQ to lock the file.
5. Call READF to read the record to be modified.
6. Modify the record and call WRITF to write it out.
7. Call POST to transfer the updated record.
8. Call RNRQ to unlock the file.

PURGE

CALL PURGE(IDCB,IERR,INAM<,<,ISC><,<,ICR>)

Delete named file INAM and all its extents, the file must not be open.

(Refer to Parameter section for description of common parameters.)

READF and EREAD

READF
CALL (IDCB,IERR,IBUF[,IL][,LEN][,NUM])
ERead

Read a record from an open file to the user buffer. If type 0 file, the number of words should be specified.

IL	Length of IBUF (read buffer), defaults are: file type = 0 Zero-length record. file type = 1 128-word record. file type > 1 Actual record length.
LEN	Actual read length, set to -1 for EOF on sequential files only.
NUM	If positive, number of record to be read; if negative, number of records to backspace. If omitted, record at current position is read. In range -32768 to 32767 for READF; -2147483648 to 2147483647 for EREAD. Applies only to file types 1 and 2.

RWPDF

CALL RWPDF(IDCB[,IERR])

Rewind a magnetic tape or position a disk file to the first record in the file.

WRITF and EWRIT

```
WRITF
CALL      (IDCB,IERR,IBUF[,IL][,NUM])
EWRIT
```

Write a record from the user's buffer to an open file. For type 0 or type 3 and above, a specified number of words is written. For type 1 and 2 files the exact record length is written.

IL Length of write buffer, defaults are:

file type = 0	Zero-length record.
file type = 1	128-word record.
file type = 2	Actual record length.
file type > 2	Zero-length length.

NUM If positive, number of record to be written; if negative, number of records to backspace. If omitted or 0, record is written to current file position. In range -32768 to 32767 for WRITF; -2147483648 to 2147483648 for EWRIT. Applies only to file types 1 and 2.

GASP

GASP Runstring	11-1
^AB	11-1
^CJ	11-1
^CS	11-2
^DA	11-2
^DJ	11-3
^DS	11-3
^EX	11-3
^KS	11-4
^RS	11-4
^SD	11-4
^SU	11-5
^UP	11-5
^??	11-5

RU,GASP[,lu]

Schedule GASP to prompt for command from LU (default = user's terminal).

RU,GASP,command

Schedule GASP, execute command, then terminate.

- lu Logical unit of interactive device on which GASP commands are entered. In a session environment LU must be specified if it is different from the session logical unit.
- command Any GASP operator command.

^ AB,job#[,u.g]

Before a job is processed, it may be removed with the AB command.

- job# Number assigned to job by spool system; use DJ to display job numbers.
- u.g Aborts all jobs owned by session account (user.group).

^ CJ,job# < ,H ,R ,priority >

Change job priority or status. Only used for a job in I, R, or RH status.

- job# Number assigned to job by spool system; use DJ to display job numbers.
- priority New job priority; only allowed before job is active.
- H Hold job from processing; changes R status to RH, and I to IH.
- R Release job for processing; changes RH status to R.

**^ CS,spoolfile< ,priority
,H >[,u.g]
,R**

Change status of outspool file or change spool priority if outspool file is not active.

spoolfile	Name of spool file as displayed by DJ.
priority	New outspool priority.
H	Hold spool file; if active, changes status to AH; if waiting, changes status to H.
R	Release spool file that has been held in AH or H status.
u.g	Change the status of spool file belonging to session account (user.group).

^ DA

Deallocate spooling. Before using DA, the spool system must be shut down, all files must be closed, and all current job processing and/or outspooling should be completed.

Only the System Manager can execute this command.

Response:

KILL SPOOLING?

The system prints this message in response to DA in order to give you a chance to change your mind.

^ DJ[AL] < ,job # [,u.g] > ,jobname

Display the job number, job name, job status, priority, user.group, and the spool files assigned to the job except for the job input spool.

- AL Report all jobs (session and non-session).
- job # Display job number of particular job.
- jobname Name of the job or jobs to be displayed. If both job # and jobname are omitted, all jobs currently in the system for the current user are displayed.
- u.g Reports only jobs belonging to the user.group account of u.g. If the “@” character is used for either the user or group, then all session users or groups (or both) are reported.

^ DS[AL][,lu[,u.g]]

Display the spool file name, job number, user.group name, outspool priority, spool status, and the LU to which the file is being or will be outspooled.

- AL Causes all spools (session and non-session) to be reported.
- lu Outspool LU; only files directed to this LU are displayed; if omitted, all files in the outspool queue are displayed. If in session, LU is the session LU, and the LU displayed is the system LU that the session LU maps to.
- u.g Reports only files belonging to the account of user.group. If the “@” character is used for either the user or group, then all users or groups (or both) are reported.

^ EX

Terminate GASP.

**^KS< ,spoolfile >[,u.g]
 ,lu**

Remove outspool file from the outspool queue.

- | | |
|-----------|--|
| spoolfile | Name of spool file to be removed. |
| lu | LU of device to which file is being outspooled. When running under session, LU is the session logical unit number. |
| u.g | Kills all spool files owned by session account user.group. |

^RS,spoolfile[,lu][,u.g]

Restart active outspool file from the beginning.

- | | |
|-----------|--|
| spoolfile | Name of active or active-held spool file in outspool queue. |
| lu | New LU to which file is to be outspooled; if omitted, LU previously assigned is used for spool output. |
| u.g | Restart spool file belonging to session account user.group. |

**^SD< ,B[ATCH] >
 ,S[POOL]**

Hold all spooled jobs, all spooled output, or both.

- | | |
|------|---|
| B | Hold all pending jobs; spool files are not affected. |
| S | Hold all pending spool files; job processing is not affected. |
| none | If both B and S are omitted, then both job processing and outspooling are held. Inspooling by JOB may continue. |

**^SU< ,B[ATCH] >
 ,S[POOL]**

Start spool system after it has been shut down with SD.

- | | |
|------|--|
| B | Jobs held with SD are released; does not restart outpooling. |
| S | Outpools held with SD are released; does not restart job processing. |
| none | Both jobs and outpools held by SD are restarted. |

^UP[,RS]

Up outpool device.

- | | |
|----|--|
| RS | Restart active files from the beginning. |
|----|--|

^??[,error#[,lu]]

- | | |
|--------|--|
| error# | Positive or negative GASP error code. Default is last error displayed. If 99 is specified, an explanation of all error codes is displayed. |
| lu | LU where the error message is displayed. |

Account Commands

Account ID Format	12-1
Global Parameters	12-1
/A	12-1
AL	12-1
/E	12-2
EX	12-2
HE	12-2
/HE	12-2
LI	12-2
LO	12-3
NE	12-3
PA	12-3
PU	12-4
RE	12-4
SD	12-4
SU	12-4
TE	12-5
TR	12-5
UN	12-5

Certain commands require system manager status. An Account Error might indicate a user has insufficient capability.

Account ID Format

USER.GROUP

- @.group All users in GROUP.
- user.@ All users named USER.
- @@ All users.

Global Parameters

namr filename:sc:cr or LU

/A

Abort current command. (Exits ACCTS if at NEXT? prompt.)

AL[TER],A[CCT]

Alter global session monitor parameters.

AL[TER],G[ROUP],<NAME>

Alter group account.

AL[TER],U[SER],<USER.GROUP>

Alter one or more attributes defined for user.

/E

End current phase. (Exits ACCTS if at NEXT? prompt.)

EX[IT]

Terminate the account setup program.

HE[LP][,<KEYWORD>][<LIST NAMR>]]

List syntax of commands.

HE[LP][,<ERROR NUMBER> [,<LU>]

Get help message for error.

/HE[LP],<ERROR NUMBER> [,<LU>]

Get help message for error from within a command.

**LI[ST],A[CCT],[<LIST NAMR>
[,<AC,PO,CO, or AL>]]**

List session wide information.

**LI[ST],G[ROUP],<NAME> [, [<LIST NAME>
[,PA],ID]]**

List one or more group account entries.

**LI[ST],[U[SER]],<USER>GROUP>[, [<LIST
NAMR>]][,PA][,ID]]**

List one or more user account entries.

LO[AD],<NAMR> [,ACCTS]

Rebuild the accounts only from an unloaded ACCT file.

LO[AD],<NAMR> [,ALL]

Rebuild the account system from an unloaded ACCT file.

LO[AD],0

Allow expansion of the accounts file.

NE[W],G[ROUP]

Create an account entry for a new group.

NE[W],U[SER]

Create an account entry for a new user.

PA[SSWORD]

Alter current user's password.

PU[RGE],A[CCT]

Purge the entire account structure.

PU[RGE],G[ROUP],<NAME>

Remove a group from the account file.

PU[RGE],U[SER],<USER.GROUP>

Remove a user from the account file.

RE[SET],G[ROUP],<NAME> [(CPU or CONNECT)]

Clear group logon clocks.

RE[SET],U[SER],<USER.GROUP> [(CPU or CONNECT)]

Clear user logon clocks.

SD,[LU[,RP][,RG]] or SD,RE[LEASE MEMORY]

Shut down an active session or the entire session system.

SU

Restart the session system.

**TE[LL], <USER.GROUP>, <MESSAGE FILE>,
<MESSAGE>**

Send a message to one or more active sessions.

**TR[[[ANSWER], <CONTROLNAMR>], <LIST
NAMR>], (EC[HO] or NO[ECHO])]**

Transfer control of ACCTS program to a file or LU from within a command.

UN[LOAD], <NAMR>

Dump a copy of the account file.

Batch and Spooling Commands

AB	13-1
CS	13-1
EOJ	13-1
JOB	13-2
SL	13-2
RUN	13-4
TL	13-4
XE	13-4

AB

Capability: 30

Terminate batch job.

CS,lu,attribute

Capability: 30

Modify or change spool options set by SL command.

lu	LU defined at setup.
attribute	One of the following:
RWind	Reset file to first record.
PURge	Change SAve flag to PURge.
SAve	Change PURge flag to SAve.
PAss	Remove HOld option.
ENd	Write EOF and terminate spool. Spool file placed in outspool queue (default).
BUffer	Change to buffering.
NBUffer	Change to no buffering.
NPass	Change LU and/or priority information, by specifying the two additional parameters: [,outlu[,priority]] outlu = new LU priority = new priority

EOJ[,RP[,RG]]

Capability: 30

End of spooled job.

RP	Dismount job's private session cartridges (default = leave mounted).
RG	Dismount job's group session cartridges (default = leave mounted).

JOB[,name[:hr:min:sec][,user[,priority [,spool priority][,sp]]]]]

Capability: 30

Initiate job for spooling.

name	Job name.
:hr:min:sec	CPU time limit for job in hours, minutes, seconds.
user	Session user account ID in the form “user.group/password”. If a job is submitted outside of a session, when session is installed this parameter must be specified.
priority	Job priority in range from 1–255 (default = 99).
spool priority	Outspool priority (default = priority 99).
sp	Specify: NO Outspool now, or NS No outspooling

SL,lu[,filedes[,attribute[,outlu[,priority [,prog]]]]]]]

Capability: 30

Spool setup and outspool control.

lu	The session LU to which a spool file is to be associated. The LU must not be LU 2 (system disk), LU 3 (auxiliary disk), any LU associated with a disk driver, a spool LU, or if in a job system LU 5 (standard spool input device).
filedes	Name of existing file to be used as a spool file (default = system assigns spool pool file).
attribute	Defines characteristics of spool access. Any three attribute codes can be combined, no delimiters necessary.

Attribute codes:

NO	Queue file for immediate outspool.
RE	Read only.
WR	Write only.
BO	Both read and write.
WN	Write now.
BU	Buffered.
PU	Purge.
SH	Write spool headers.
ST	Standard file codes.

Default for attribute codes:

	outlu specified	outlu not specified
filedes specified	WRITE,HOLD, SPOOL HEADERS, SAVE	READ,HOLD, STANDARD FORMAT SAVE
filedes not specified	WRITE,HOLD SPOOL HEADERS, SPOOL POOL FILE,PURGE	BOTH,HOLD, STANDARD FORMAT, SPOOL POOL FILE,PURGE

outlu	Session LU for outspooling.
priority	Outspool priority (default = session – 99, batch – priority of job).
prog	If specified, program “prog” will be scheduled, with wait, by the spool system when spool LU is closed. Note the spool file will not be outspooled; “prog” must properly dispose of the file. Required capability of 50.

RUN,JOB,namr[,priority]

Capability: 30

Run batch job.

- | | |
|----------|---|
| namr | File name of file containing single job to be spooled, or LU of input device containing jobs to be spooled (default = session terminal, or LU 5 if outside of session). |
| priority | Priority of job. If omitted, priority for job is taken from the job statement. If a job priority is not specified in the job statement, default is 99. |

TL:hr:min:sec

Capability: 30

Set run time limit.

- | | |
|-------------|--|
| :hr:min:sec | Time limit for execution of any programs with RU command subsequent to TL command. If omitted, job time limit is used. |
|-------------|--|

XE,namr[,priority]

Capability: 30

Job input control.

- | | |
|----------|--|
| namr | Identifies input device containing a job to be placed in job queue, may be a logical unit or the name of an existing file. |
| priority | Job priority (default = 99). |

SMP Calls

Parameters	14-1
SOPN	14-1
Working Calls	14-2
Retrieve Record Position	14-2
Change Record Position	14-2

Parameters

- ISMP Three-word array containing name of program SMP.
- ISLU Spool LU returned by SPOPEN call. Each subsequent spool call must specify this LU.

SPOPEN

CALL SPOPEN(IBUFR,ISLU)

Make a spool file active and ready for use.

IBUFR 16-word setup buffer structured as follows:

Word	Contents
0	=0 if no batch input checking desired.
1	>0 session LU for the spool file; or =0 SMP allocates a session LU for spool file; or =-1 a direct map to system LU is set up.
2,3,4	Six-character file name of user file used for spooling. Cannot be a spool pool file.
5	Security code.
6	Cartridge Reference Number (CRN).
7	Driver type, in octal.
8	Disposition flags:

15	14	13 . . . 9	8	7	6	5	4	3	2	1	0
BU	BI		W/R				ST	SP		HO	SA

- BU 1= buffered; 0= not buffered.
- BI 1= batch input; 0 otherwise.
- W/R 10B= write; 01B= read; 00B= write/read.
- ST 1= standard file; 0= spool file.
- SP 1= spool pool file; 0= user file.
- HO 1= hold outspool; 0= outspool now.
- SA 1= save file; 0= purge.
- 9 Spool priority (1-9999).
- 10 Spool status (used by SMP, GASP).
- 11 If batch (job number); if not batch (directory entry number of session program).
- 12-14 Set to 0 or program parameter of SL command.
- 15 Outspool LU.

ISLU Spool LU return.

Working Calls

CALL EXEC(23,ISMP,XX,ISLU)

- XX =1 Change purge to save.
- =2 Change save to purge.
- =3 Queue for outspooling.
- =4 EOF and queue for outspooling.
- =5 Change spool options; use additional parameters NULU and NPR following ISLU for this call only: NULU New outspool LU (default = previous LU). NPR New outspool priority (default = previous value).
- =6 Set buffer flag.
- =7 Clear buffer flag.
- =8 Retrieve record position.
- =9 Change record position.

Retrieve Record Position

CALL EXEC(23,ISMP,8,ISLU)

CALL RMPAR(IPRM) (for parameter pickup)

- IPRM 5-word array containing pointers to record position.
- | | | |
|--------|---|---|
| word 1 | } | Contains an internal coding of the current position of the referenced file. |
| word 2 | | |
| word 3 | | |
| word 4 | } | Not used but should be included in array. |
| word 5 | | |

Change Record Position

CALL EXEC(23,ISMP,9,ISLU,IPRM1,IPRM2,IPRM3)

- IPRM1-3 Record position from the RETRIEVE RECORD call.

VMA/EMA Routines

General Purpose VMA/EMA Subroutines	15-1
EIOSZ	15-1
EMAST	15-1
LKEMA/ULEMA	15-1
VMAIO	15-2
VMAST	15-2
VMA Backing Store File Subroutines	15-3
VMACLOSE	15-3
VMAOPEN	15-3
VMAPOST	15-4
VMAPURGE	15-4
VMAREAD	15-4
VMAWRITE	15-4
FMGR VMA File Routines	15-5
CLSVM	15-5
CREVM	15-5
OPNVM	15-6
PSTVM	15-6
PURVM	15-6
VREAD	15-7
VWRIT	15-7
VMA/EMA Mapping Management Subroutines	15-8
.EMIO Subroutine	15-8
.ESEG Subroutine	15-9
.IMAP Subroutine	15-10
.IRES Subroutine	15-11
.JMAP Subroutine	15-12
.JRES Subroutine	15-13
.LBP, .LBPR Subroutine	15-14
.LPX, .LPXR Subroutine	15-14
MMAP Subroutine	15-15

General Purpose VMA/EMA Subroutines

EIOSZ

Purpose: Determine maximum length of transfer.

Syntax: CALL EIOSZ(isize)
isize = EIOSZ ()

isize Returns maximum number of words that can be used in a VMAIO call.

EMAST

Purpose: Return information about VMA or EMA.

Syntax: CALL EMAST(nema, nmseg, imseg[, iws])

nema Total page size of VMA or EMA (not including page table).

nmseg Total page size of mapping segment (MSEG), excluding spillover page.

imseg Starting logical page of MSEG.

iws Working set page size. For EMA, same as nema.

Returns:

A-Register = 0 if normal return
 = -1 if error occurred

LKEMA/ULEMA

Purpose: Lock/unlock a shareable EMA partition.

Syntax: CALL LKEMA lock partition
 CALL ULEMA unlock partition

VMAIO

<i>Purpose:</i>	Perform VMA or EMA data transfers to or from a device.
<i>Syntax:</i>	CALL VMAIO(ecode,cntrl,ibuff,ilen[,pram3[,pram4]])
ecode	I/O request code; 1 for Read, 2 for Write.
cntrl	Two-word quantity with the following format: Word 1: Bit 15 OV bit 14 Reserved 13 0 12-8 Reserved 7-0 LU Word 2: Bit 15 Not used. 14 UB Do not buffer. 13 Not used. 12 Z IPRM1 is buffer of length IPRM2 to output first (always an output regardless of ICODE). 11 Not used. 10 X Device dependent; usually honest mode. 9 A Device dependent. 8 K Device dependent on terminal echo characters. 7 V Device dependent on line printers; do not use code 1 as carriage control. 6 M Device dependent; usually binary mode. 5-0 Reserved
ibuff	Two-word integer that specifies the VMA/EMA word offset of the buffer.
ilen	Integer that specifies the length of the buffer in words (positive value) or bytes (negative value).
pram3	Optional parameter or buffer, as in EXEC 1 or 2 call.
pram4	Optional parameter of buffer length.

VMAST

<i>Purpose:</i>	Return size of VMA or EMA.
<i>Syntax:</i>	CALL VMAST (ivma, isize)
ivma	-2 = Not VMA or EMA program, isize = 0 0 = EMA program 1 = VMA program
isize	VMA or EMA page size.

VMA Backing Store File Subroutines

Default file names:

- a) on the FMGR cartridge: nnncVM
- b) on the CI volume: VMcnnn.VMA

where: nnn = ID segment number of the VMA program
 c = CPU number

The default backing store is a type 2 file in 256-block increments.

If the system generation or FMGR VL command specifies \$SCRN as zero, the default backing store file goes either on the /SRATCH/ directory (if it exists) or on the current working directory.

If \$SCRN specifies other than zero, then the backing store file goes on the specified FMGR cartridge.

VMACLOSE

Purpose: Close the VMA backing store file.

Syntax: CALL VMACLOSE

VMAOPEN

Purpose: Create or open backing store file.

Syntax: CALL VMAOPEN (ierr,name,ioptn)

ierr Integer returning error code (0 indicates successful call).

name Character string indicating file pathname (63 or fewer characters), DS information is ignored; name may be blank.

ioptn Character string indicating file options; character string list of one-letter options (upper or lowercase) selected from the following set:

- R = Open for reading.
- W = Open for writing.
- O = OK to open an existing backing store file.
- C = OK to create a new backing store file.
- S = Open shared.
- U = Open in update mode.
- X = OK to access and/or create extents.
- T = File is temporary.
- V = Defer create/open until required.

VMAPOST

Purpose: Post working set to disk.

Syntax: CALL VMAPOST

VMAPURGE

Purpose: Purge VMA backing store file.

Syntax: CALL VMAPURGE

VMAREAD

Purpose: Read data from a file to a VMA/EMA.

Syntax: `ilen = VMAREAD(idcb,ierr,iarray,idl)`

- `idcb` Data Control Block (DCB). Previously specified integer array of $144+n$ words; n is zero or positive.
- `ierr` Integer returning FMP error code. Zero indicates successful call.
- `iarray` Two-word address specifying data transfer destination start address in VMA/EMA.
- `idl` Integer specifying data length requested, in bytes.
- `ilen` Integer returning actual data length read (in bytes) or negative error code. Also negative if more than 32767 bytes were read.

VMAWRITE

Purpose: Write data from VMA/EMA to a file.

Syntax: `ilen = VMAWRITE(idcb,ierr,iarray,idl)`

- `idcb` Data Control Block (DCB). Previously specified integer array of $144 + n$ words; n is zero or positive.
- `ierr` Integer returning FMP error code. Zero indicates successful call.
- `iarray` Two-word address specifying data transfer destination start address in VMA/EMA (positive).
- `idl` Integer specifying data length requested, in bytes.
- `ilen` Integer returning actual number of bytes transferred, or negative error code. Also negative if more than 32767 bytes are transferred.

FMGR VMA File Routines

CLSVM

Purpose: Close the VMA backing store file.

Syntax: Call CLSVM

CREVM

Purpose: Create a VMA backing store file.

Syntax: CALL CREVM(name,ierr,ioptn,isc,icr)

name Three-word array returning file name.

ierr Integer returning error code. Zero indicates successful call.

ioptn File options:

Bit 0 = 0 Ignore name and bit 1.

Bit 0 = 1 Create non-scratch file for backing store file.

Bit 1 = 1 Open file if create fails due to duplicate file error.

Bit 2 = 1 Defer file create until working set must be written to file.

Bit 3 = 1 Do not create or address file extents.

isc Integer specifying file security code.

icr Integer specifying file Cartridge Reference Number (CRN).

OPNVM

<i>Purpose:</i>	Open a VMA backing store file.
<i>Syntax:</i>	CALL OPNVM (name,ierr,ioptn,isc,icr)
name	Three-word array returning file name.
ierr	Integer returning error code. Zero indicates successful call.
ioptn	File options: Bit 0 = 1 Open file for nonexclusive use. Bit 1 = 1 Open file for update. Bit 2 = 1 Defer file open until required. Bit 3 = 1 Do not address or create file extents. Bit 4 = 1 Open disk file for read-only access
isc	Integer specifying file security code.
icr	Integer specifying file Cartridge Reference Number (CRN).

PSTVM

<i>Purpose:</i>	Post working set to disk.
<i>Syntax:</i>	CALL PSTVM

PURVM

<i>Purpose:</i>	Purge VMA backing store file.
<i>Syntax:</i>	CALL PURVM

VREAD

<i>Purpose:</i>	Read data from a file to a VMA/EMA.
<i>Syntax:</i>	CALL VREAD(idcb,ierr,iarray,idl[,ilen[,inum]])
idcb	Data Control Block (DCB). Previously specified integer array of $144+n$ words; n is zero or positive.
ierr	Integer returning error code. Zero indicates successful call.
iarray	Two-word integer representing data transfer destination start address in VMA/EMA (positive).
idl	Integer specifying data length requested in words (positive).
ilen	Integer indicating actual data length read.
inum	Integer specifying positive offset (in records) from beginning of file or negative offset from current position. Default = current position.

VWRIT

<i>Purpose:</i>	Write data from VMA/EMA to a file.
<i>Syntax:</i>	CALL VWRIT(idcb,ierr,iarray,idl[,inum])
idcb	Data Control Block (DCB). Previously specified integer array of $144+n$ words; n is zero or positive.
ierr	Integer returning error code. Zero indicates successful call.
iarray	Two-word integer representing data transfer destination start address in VMA/EMA (positive).
idl	Integer specifying data length requested, in words (positive).
inum	Integer specifying positive record number or negative number of records to backspace (file types 1 or 2 only). Default is record at current position.

VMA/EMA Mapping Management Subroutines

.EMIO Subroutine

Purpose: Map in up to MSEG-size buffer, which can then be used for I/O.

Macro/1000 calling sequence:

```
EXT .EMIO
JSB .EMIO
DEF RTN      Address for error return
DEF BUFL    Number of words in the buffer
DEF TABLE  Table containing array parameters
DEF An      Subscript value for nth dimension
           :
DEF A1      Subscript value for 1st dimension
```

RTN error return

Normal return:

B-Register contains logical address of element.
A-Register is meaningless.

Table:

```
Number of Dimensions
-L(n)
d(n-1)
:
-L(2)
d(1)
-L(1)
Number of words per element
Offset word 1 (bits 15-0)
Offset word 2 (bits 31-16)
```

where:

$L(i)$ is the lower bound of the i th dimension.

$d(i)$ is the number of elements in the i th dimension.

.ESEG Subroutine

Purpose: Map several pages of EMA or VMA (not necessarily contiguous) into logical memory.

Macro/1000 calling sequence:

	EXT .ESEG	
	:	
	LDB number	Number of map registers to modify.
	JSB .ESEG	
	DEF *+2	Error return point (not used).
	DEF PBUFR	Table of pages to map.
RTN	error return	(Not used).
RTN+1	normal return	Normal return point.

Normal return:

All VMA/EMA pages are mapped into logical memory and B-Register equals the logical address of the starting page of MSEG.

.IMAP Subroutine

Purpose: Resolve address of array element with one-word integer subscripts and map it into logical memory.

Macro/1000 calling sequence:

EXT	.IMAP	
JSB	.IMAP	
DEF	TABLE	Address of table containing array parameters.
DEF	A_n	Address of nth subscript value.
	:	:
DEF	A1	Address of 1st subscript value.
RTN	normal return	

Normal return:

B-Register contains logical address of the element.
A-Register is undefined.

Table:

DEC		Number of dimensions.
DEC	$D(n-1)$	Number of elements in the $(n-1)$ dimension.
DEC	$D(n-2)$.
	:	:
DEC	$D(1)$	Number of elements in the first dimension.
DEC		Number words per element.
BSS	2	Offset word in double integer format (most significant word first, least significant word last).

.IRES Subroutine

Purpose: Resolve address of array element with one-word integer subscripts.

Macro/1000 calling sequence:

```
EXT .IRES
JSB .IRES
DEF TABLE Address of table containing array
              parameters.
DEF An      Address of nth subscript value.
  :
  :
DEF A1      Address of 1st subscript value.
RTN        normal return
```

Normal return:

B-Register contains logical address of the element.
A-Register is undefined.

Table:

```
DEC          Number of dimensions.
DEC D(n-1)   Number of elements in the (n-1)
              dimension.
DEC D(n-2)   .
  :
  :
DEC D(1)     Number of elements in the first
              dimension.
DEC          Number of words per element.
BSS 2       Offset word in double integer format
              (most significant word first, least
              significant word last).
```

.JMAP Subroutine

Purpose: Resolve address of array element with double integer subscripts and map it into logical memory.

Macro/1000 calling sequence:

```
EXT .JMAP
JSB .JMAP
DEF TABLE Address of the array description table.
DEF An Address of  $n$ th subscript value.
DEF A ( $n-1$ ) Address of ( $n-1$ ) subscript value.
:
:
DEF A1 Address of 1st subscript value.
RTN normal return
```

Normal return:

VMA or EMA array element resides in physical memory, last two user map registers (VSEG) point to that element, and B-Register contains logical address of element. A-Register is undefined.

Table:

```
DEC Number of dimensions.
DEC D ( $n-1$ ) Number of elements in the ( $n-1$ )
dimension (High Bits).
DEC D ( $n-1$ ) Number of elements in the ( $n-1$ )
dimension (Low Bits).
:
:
DEC D (1) Number of elements in the first
dimension (High Bits).
DEC D (1) Number of elements in the first
dimension (Low Bits)
DEC Number of words per element.
BSS 2 Offset word in double integer
format.
```

.JRES Subroutine

Purpose: Resolve address of array element with double integer subscripts.

Macro/1000 calling sequence:

```
EXT .JRES
JSB .JRES
DEF TABLE Address of table containing array
           parameters.
DEF An     Address of nth subscript value.
           :
           :
DEF A1     Address of 1st subscript value.
RTN       normal return
```

Normal return:

A- and B-Registers contain offset of array element into VMA or EMA in double integer format (most significant word in A-Register, least significant word in B-Register).

Table:

```
DEC      Number of dimensions.
DEC D(n-1) Number of elements in the (n-1)
           dimension (High Bits).
DEC D(n-1) Number of elements in the (n-1)
           dimension (Low Bits).
           :
           :
DEC D(1)   Number of elements in the first
           dimension (High Bits).
DEC D(1)   Number of elements in the first
           dimension (Low Bits).
DEC      Number of words per element
BSS 2     Offset word in double integer
           format.
```

.LBP, .LBPR Subroutine

Purpose: Convert virtual address to logical address.

Macro/1000 calling sequence:

```
EXT .LBP          EXT .LBPR
DLD PONTR        or  JSB .LBPR
JSB .LBP         DEF PONTR
```

where:

PONTR Double integer pointer (high word first) containing virtual address.

Normal return:

B-Register contains logical address.
A-Register contains data's page number in physical memory.

.LPX, .LPXR Subroutine

Purpose: Convert virtual address of offset to logical address.

Macro/1000 calling sequence:

```
EXT .LPX          EXT .LPXR
DLD PONTR        or  JSB .LPXR
JSB .LPX         DEF PONTR
DEF OFFSET       DEF OFFSET
```

where:

PONTR Double integer pointer containing the virtual address.

OFFSET Double integer offset from the virtual address.

Normal return:

B-Register contains logical address.
A-Register contains data's page number in physical memory.

MMAP Subroutine

Purpose: Map consecutive pages of EMA or VMA into logical memory.

FORTTRAN calling sequence:

CALL MMAP(ipgs,npgs)

ipgs VMA/EMA page holding start of buffer to map (where first page in EMA or VMA is page 0).

npgs Number of pages (rounded up) in buffer to be mapped.

Macro/1000 calling sequence:

```
EXT MMAP
JSB MMAP
DEF RTN
DEF IPGS
DEF NPGS
```

RTN return point

Upon return:

A-Register = 0 if normal return.
 = -1 if an error occurred.

Tables

ASCII Characters and Octal Codes	16-1
ASCII Character Set in the Two-Byte Word	16-2
ASCII Character Set	16-3
RTE Special Characters	16-7
Instruction Codes in Octal	16-8
Base Set Instruction Codes in Binary	16-10
Extended Instruction Group Codes in Binary	16-11
System Communications Area Locations	16-14
Device Reference Table (DRT)	16-16
Equipment Table (EQT)	16-16
Legend for EQT Table	16-17
Device Status Table A	16-18
Device Status Table B	16-19
Device Status Table Key	16-20
EQT Word 6	16-21
ID Segment	16-22
ID Segment (continued)	16-23
ID Segment Extension	16-24
Session Control Block (SCB)	16-25
System Disk Layout	16-26
Device File DCB	16-27
Data Control Block (DCB) Format	16-28
Legend For Data Control Block	16-29
FMGR Cartridge Specification Entry	16-30
FMGR Directory File Entry	16-31
FMGR Type 0 File Directory Entry	16-32
Disk File Record Formats	16-33
Type 6 File Format	16-34
CI Volume Header Format	16-35
CI Directory Flag Word Format	16-35
CI Directory Structure	16-36
CI Root Directory Header/Trailer	16-37
CI Root Directory Entry	16-37
CI Directory Header/Trailer Format	16-38
CI File Entry	16-38
CI Subdirectory Entry	16-39
CI Extent Entry	16-39
NAM Record	16-40

Extended NAM Record	16-41
ENT Record	16-42
Extended ENT Record (XENT)	16-42
EXT Record	16-43
Extended EXT Record (XEXT)	16-43
DBL Record	16-44
Extended DBL Record (XDBL)	16-45
EMA Record	16-46
END Record	16-46
Extended END Record (XEND)	16-47
GEN Record	16-47
LOD Record	16-48
Data Record	16-48
RPL Record	16-49
Allocate Record	16-49
MSEG Record	16-50
Absolute Tape Format	16-50
FMGR Global Equivalence	16-51
General Wait State Messages (State 3)	16-52
Cartridge Directory Format (Cartridge List)	16-53

ASCII Characters and Octal Codes

	00x	01x	02x	03x	04x	05x	06x	07x	10x	11x	12x	13x	14x	15x	16x	17x
xx0	Nul	Bs	Dle	Can	Sp	(0	8	@	H	P	X	'	h	p	x
xx1	Soh	Ht	Dc1	Em	!)	1	9	A	I	Q	Y	a	i	q	y
xx2	Stx	Lf	Dc2	Sub	"	*	2	:	B	J	R	Z	b	j	r	z
xx3	EtX	Vt	Dc3	Esc	#	+	3	;	C	K	S	[c	k	s	{
xx4	Eot	Ff	Dc4	Fs	\$,	4	<	D	L	T	\	d	l	t	
xx5	Enq	Cr	Nak	Gs	%	-	5	=	E	M	U]	e	m	u	}
xx6	Ack	So	Syn	Rs	&	.	6	>	F	N	V	^	f	n	v	~
xx7	Bel	Si	EtB	Us	'	/	7	?	G	O	W	_	g	o	w	Del

* Pressing the Control Key while typing an uppercase letter produces the corresponding control code on most terminals. For example, Control-H is a backspace.

ASCII Character Set in the Two-Byte Word

		000-037B		040-077B		100-137B		140-177B										
BITS		column	column	column	column	column	column	column	column									
row		0	1	2	3	4	5	6	7									
b ₈	b ₇	0	0	0	0	0	0	0	0									
b ₆	b ₅	0	0	0	0	1	1	1	1									
b ₄	b ₃	0	1	2	3	4	5	6	7									
b ₂	b ₁	0	1	2	3	4	5	6	7									
0	0	0	0	0	0	0	0	0	0	0	NUL	DLE	SP	0	@	P	\	p
0	0	0	0	1	1	1	1	1	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	0	0	0	0	0	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	0	0	0	0	0	0	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	0	0	0	0	0	0	4	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	0	0	0	0	0	0	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	0	0	0	0	0	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	0	0	0	0	0	0	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	0	0	0	0	0	0	8	BS	CAN	(8	H	X	h	x
1	0	0	1	0	0	0	0	0	0	9	HT	EM)	9	I	Y	i	y
1	0	1	0	0	0	0	0	0	0	10	LF	SUB	▪	:	J	Z	j	z
1	0	1	1	0	0	0	0	0	0	11	VT	ESC	+	;	K	l	k	{
1	1	0	0	0	0	0	0	0	0	12	FF	FS	,	<	L	\	l	
1	1	0	1	0	0	0	0	0	0	13	CR	GS	-	=	M]	m	}
1	1	1	0	0	0	0	0	0	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	0	0	0	0	0	0	15	SI	US	/	?	O	_	o	DEL

0

32 CONTROL CODES

64 CHARACTER SET

96 CHARACTER SET

128 CHARACTER SET

Unshifted Lower Case

Example: The representation for the character "K" (column 4, row 11) is:

	b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
BINARY	0	1	0	0	1	0	1	1
OCTAL	1		1		3			

* Depressing the Control key while typing an upper case letter produces the corresponding control code on most terminals. For example, Control-H is a backspace.

ASCII Character Set

Decimal Value	Octal Values		Mnemonic	Graphic	Meaning
	Left Byte	Right Byte			
0	000000	000000	NUL	N U	Null
1	000400	000001	SOH	S H	Start of Heading
2	001000	000002	STX	S X	Start of Text
3	001400	000003	ETX	E X	End of Text
4	002000	000004	EOT	E T	End of Transmission
5	002400	000005	ENG	E Q	Enquiry
6	003000	000006	ACK	A K	Acknowledge
7	003400	000007	BEL		0000007
8	004000	000010	BS	B S	Backspace
9	004400	000011	HT	H T	Horizontal Tabulation
10	005000	000012	LF	L F	Line Feed
11	005400	000013	VT	V T	Vertical Tabulation
12	006000	000014	FF	F F	Form Feed
13	006400	000015	CR	C R	Carriage Return
14	007000	000016	SO	S O	Shift Out
15	007400	000017	SI	S I	Shift In } Alternate Character Set
16	010000	000020	DLE	D L	Data Link Escape
17	010400	000021	DC1	D 1	Device Control 1 (XON)
18	011000	000022	DC2	D 2	Device Control 2 (TAPE)
19	011400	000023	DC3	D 3	Device Control 3 (XOFF)
20	012000	000024	DC4	D 4	Device Control 4 (TAPE)
21	124000	000025	NAK	N K	Negative Acknowledge
22	013000	000026	SYN	S Y	Synchronous Idle
23	013400	000027	ETB	E B	End of Transmission Block
24	014000	000030	CAN	C N	Cancel
25	014400	000031	EM	E M	End of Medium
26	015400	000032	SUB	S B	Substitute
27	015400	000033	ESC	E C	Escape
28	016000	000034	FS	F S	File Separator
29	016400	000035	GS	G S	Group Separator
30	017000	000036	RS	R S	000036
31	017400	000037	US	U S	000037
127	077400	000177	DEL	■	Delete, Rubout

ASCII Character Set (continued)

Decimal Value	Octal Values		Character	Meaning
	Left Byte	Right Byte		
32	020000	000040		Space, blank
33	020400	000041	!	Exclamation Point
34	021000	000042	"	Quotation Mark
35	021400	000043	#	Number Sign, Pound Sign
36	022000	000044	\$	Dollar Sign
37	022400	000045	%	Percent
38	023000	000046	&	Ampersand, And Sign
39	023400	000047	'	Apostrophe, Acute Accent
40	024000	000050	(Left (opening) Parenthesis
41	004400	000051)	Right (closing) Parenthesis
42	025000	000052	*	Asterisk, Star
43	025400	000053	+	Plus
44	026000	000054	,	Comma, Cedilla
45	026400	000055	-	Hyphen, Minus, Dash
46	027000	000056	.	Period, Decimal Point
47	027400	000057	/	Slash, Slant
48	030000	000060	0	} Digits, Numbers
49	030400	000061	1	
50	031000	000062	2	
51	031400	000063	3	
52	032000	000064	4	
53	032400	000065	5	
54	033000	000066	6	
55	033400	000067	7	
56	034000	000070	8	
57	034400	000071	9	
58	035000	000072	:	Colon
59	035400	000073	;	Semicolon
60	036000	000074	<	Less Than
61	036400	000075	=	Equals
62	037000	000076	>	Greater Than
63	037400	000077	?	Question Mark
64	040000	000100	@	Commercial At

ASCII Character Set (continued)

Decimal Value	Octal Values		Character	Meaning
	Left Byte	Right Byte		
65	040400	000101	A	Uppercase Alphabet. Capital Letters
66	041000	000102	B	
67	041400	000103	C	
68	042000	000104	D	
69	042400	000105	E	
70	043000	000106	F	
71	043400	000107	G	
72	044000	000110	H	
73	044400	000111	I	
74	044400	000112	J	
75	045400	000113	K	
76	046000	000114	L	
77	046400	000115	M	
78	047000	000116	N	
79	047400	000117	O	
80	050000	000120	P	
81	050400	000121	Q	
82	051000	000122	R	
83	051400	000123	S	
84	052000	000124	T	
85	052400	000125	U	
86	053000	000126	V	
87	053400	000127	W	
88	054000	000130	X	
89	054400	000131	Y	
90	055000	000132	Z	
91	055400	000133	[Left (opening) Bracket
92	056000	000134	\	Backslash, Reverse Slant
93	056400	000135]	Right (closing) Bracket
94	057000	000136	^	Caret, Circumflex
95	057400	000137	_	Underline
96	060000	000140	`	Grave Accent

ASCII Character Set (continued)

Decimal Value	Octal Values		Character	Meaning
	Left Byte	Right Byte		
97	060400	000141	a	Lowercase Letters
98	061000	000142	b	
99	061400	000143	c	
100	062000	000144	d	
101	062400	000145	e	
102	063000	000146	f	
103	063400	000147	g	
104	064000	000150	h	
105	064400	000151	i	
106	065000	000152	j	
107	065400	000153	k	
108	066000	000154	l	
109	066400	000155	m	
110	067000	000156	n	
111	067400	000157	o	
112	070000	000160	p	
113	070400	000161	q	
114	071000	000162	r	
115	071400	000163	s	
116	072000	000164	t	
117	072400	000165	u	
118	073000	000166	v	
119	073400	000167	w	
120	074000	000170	x	
121	074400	000171	y	
122	075000	000172	z	
123	075400	000173	{	Left (Opening) Brace
124	076000	000174	:	Vertical Line
125	076400	000175	}	Right (closing) Brace
126	077000	000176	~	Tilde, Overline

RTE Special Characters

Mnemonic	Octal Value	Use
SOH (Control-A)	1	Backspace (TTY)
EM (Control-Y)	31	Backspace (HP 2600)
BS (Control-H)	10	Backspace (TTY, HP 2615, 2640, 2644, 2645)
EOT (Control-D)	4	End-of-file (TTY, HP 2625, 2640, 2644, 2645)

Instruction Codes in Octal

<p>Memory Reference</p> <p>ADA 04(0XX)...</p> <p>ADB 04(1XX)...</p> <p>ADN 01(0XX)...</p> <p>CPA 05(0XX)...</p> <p>CPB 05(1XX)...</p> <p>IOR 03(0XX)...</p> <p>ISZ 03(1XX)...</p> <p>JMP 02(1XX)...</p> <p>JSB 01(1XX)...</p> <p>LDA 06(0XX)...</p> <p>LDB 06(1XX)...</p> <p>STA 07(0XX)...</p> <p>STB 07(1XX)...</p> <p>XOR 02(0XX)...</p> <p style="text-align: center;">↑ Binary</p> <p>Shift-Rotate</p> <p>ALF 001700</p> <p>ALR 001400</p> <p>ALS 001000</p> <p>ARS 001100</p> <p>BLF 005700</p> <p>BLR 005400</p> <p>BLS 005100</p> <p>CLE 000040</p> <p>ELA 001600</p> <p>ELB 005600</p> <p>ERA 001500</p> <p>ERB 005500</p> <p>NOP 000000</p> <p>RAL 001200</p> <p>RAR 001300</p> <p>RBL 005200</p> <p>RBR 005300</p> <p>SLA 000010</p> <p>SLB 004010</p> <p>Alter-Skip</p> <p>CCA 003400</p> <p>CCB 007400</p> <p>CCE 002300</p> <p>CLA 002400</p> <p>CLB 006400</p> <p>CLE 002100</p> <p>CMA 003000</p> <p>CMB 007000</p> <p>CME 002200</p> <p>INA 002004</p>	<p>Alter-Skip</p> <p>INB 006004</p> <p>RSS 002001</p> <p>SEZ 002040</p> <p>SLA 002010</p> <p>SLB 006010</p> <p>SSA 002020</p> <p>SSB 006020</p> <p>SZA 002002</p> <p>SZB 006002</p> <p>Input/Output</p> <p>CLC 1067...</p> <p>CLF 1031...</p> <p>CLO 103101</p> <p>HLT 1020...</p> <p>LIA 1025...</p> <p>LIB 1065...</p> <p>MIA 1024...</p> <p>MIB 1064...</p> <p>OTA 1026...</p> <p>OTB 1066...</p> <p>SFC 1022...</p> <p>SFS 1023...</p> <p>SOC 102201</p> <p>SOS 102301</p> <p>STC 1027...</p> <p>STF 1021...</p> <p>STO 102101</p> <p>Extended Arithmetic</p> <p>ASL 1000(09X).</p> <p>ASR 1010(01X).</p> <p>DIV 100400</p> <p>DLD 104200</p> <p>DST 104400</p> <p>LSL 1000(10X)</p> <p>LSR 1010(10X)</p> <p>MPY 100200</p> <p>RRL 1001(00X).</p> <p>RRR 1011(00X).</p> <p style="text-align: center;">↑ Binary</p>	<p>Extended Inst. Group</p> <p>ADX 105746</p> <p>ADY 105756</p> <p>CAX 101741</p> <p>CAY 101751</p> <p>CBS 105774</p> <p>CBT 105766</p> <p>CBX 105741</p> <p>CBY 105751</p> <p>CMW 105776</p> <p>CXA 101744</p> <p>CXB 105744</p> <p>CYA 101754</p> <p>CYB 105754</p> <p>DSX 105761</p> <p>DSY 105771</p> <p>ISX 105760</p> <p>ISY 105770</p> <p>JLY 105762</p> <p>JPY 105772</p> <p>LAX 101741</p> <p>LAY 101752</p> <p>LBT 105763</p> <p>LDY 105755</p> <p>MBT 105765</p> <p>MWV 105777</p> <p>SAX 101740</p> <p>SAY 101750</p> <p>SBS 105733</p> <p>SBT 105764</p> <p>SBX 105740</p> <p>SBY 105750</p> <p>SFB 105767</p> <p>STX 105743</p> <p>STY 105753</p> <p>XAX 101747</p> <p>XAY 101757</p> <p>SBS 105747</p> <p>XBY 105757</p>
--	--	--

Instruction Codes in Octal (continued)

<p>Floating Point</p> <p>FAD 105000 FDV 105060 FIX 105100 FLT 105120 FMP 105040 FSB 105020 .FIXD 105104 .FLTD 105124 .TADD 105022 .TDIV 105062 .TFTD 105126 .TFTS 105122 .TFXD 105106 .TFXS 105102 .TMPY 105042 .TSUB 105022 .XADD 105001 .XDIV 105061 .XFTD 105125 .XFTS 105121 .XFXD 105105 .XFXS 105101 .XMPY 105041 .XSUB 105021</p> <p>Scientific Inst. Set</p> <p>ALOG 105322 ALOFT 105327 ATAN 105323 COS 105324 EXP 105326 SIN 105325 SQRT 105321 TAN 105320 TANH 105330 DPOLY 105331 /CMRT 105332 /ATLG 105333 .FPWR 105334 .TPWR 105335</p>	<p>Fast FORTRAN</p> <p>DBLE 105201 DDINT 105217 SNGL 105202 .BLE 105207 .CFER 105231 .DFER 105205 .ENTP 105223 .ENTR 105223 .FLUN 105226 .GOTO 105221 .NGL 105214 .PACK 105230 .PWR2 105225 \$SETP 105227 .XCOM 105215 .XFER 105220 .XPAK 105206 .DCM 105216 .FCM 105232 .MAP 105222 .TCM 105233</p> <p>Extended Arithmetic</p> <p>.DAD 105014 .DCO 105204 .DDE 105211 .DDI 105074 .DDIR 105134 .DDS 105213 .DIN 105210 .DIS 105212 .DMP 105054 .DNG 105203 .DSB 105034 .DSBR 105114</p>	<p>Dynamic Mapping System</p> <p>DJP 105732 DJS 105733 JRS 105715 LFA 101727 LFB 105727 MBF 105703 MBI 105702 MBW 105704 MWF 105706 MWI 105705 MWW 105707 PAA 101712 PAB 105712 PBA 101713 PBB 105713 RSA 101730 RSB 105730 RVA 101731 RVB 105731 SJP 105734 SJS 105737 SSM 105714 SYA 101710 SYB 105710 UJP 105736 UJS 105737 USA 101711 USB 105711 XCA 101726 SLB 105724 SMA 101722 SMB 105722 SMB 105722 SMM 105722 SMM 105721 XSA 101725 XSB 105725</p>
---	--	--

Base Set Instruction Codes in Binary

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
D/I	AND	001	0	Z/C	← Memory Address →													
D/I	XOR	010	0	Z/C														
D/I	IOR	011	0	Z/C														
D/I	JSB	001	1	Z/C														
D/I	JMP	010	1	Z/C														
D/I	ISZ	011	1	Z/C														
D/I	AD*	100	A/B	Z/C														
D/I	CP*	101	A/B	Z/C														
D/I	LD*	110	A/B	Z/C														
D/I	ST*	111	A/B	Z/C														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0	SRG	000	A/B	0	D/E	*LS	000	†CLE	D/E	‡SL	*LS	000						
			A/B	0	D/E	*RS	001		D/E		*RS	001						
			A/B	0	D/E	R*L	010		D/E		R*L	010						
			A/B	0	D/E	R*R	011		D/E		R*R	011						
			A/B	0	D/E	*LR	100		D/E		*LR	100						
			A/B	0	D/E	ER*	101		D/E		ER*	101						
			A/B	0	D/E	EL*	110		D/E		EL*	110						
			A/B	0	D/E	*LF	111		D/E		*LF	111						
			A/B	0	0	L*E	101		0		L*E	101						
			A/B	0	0	S*E	110		0		S*E	110						
				000		NOF	000		000		NOF	000						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0	ASG	000	A/B	1	CL*	01	CLE	01	SEZ	SS*	SL*	IN*	SZ*	RSS				
			A/B		CM*	10	CME	10										
			A/B		CC*	11	CCE	11										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
1	IOG	000		1	H/C	HLT	000	← Select Code →										
				1	0	STF	001											
				1	1	CLF	001											
				1	0	SFC	010											
				1	0	SFS	011											
			A/B	1	H/C	MI*	100											
			A/B	1	H/C	LI*	101											
			A/B0	1	H/C	OT*	110											
			1	1	H/C	STS	111											
				1	H/C	CLC	111											
				1	0	STO	001											
				1	1	CLO	001											
				1	H/C	SOC	010											
				1	H/C	SOS	011											
								000		001								
								000		001								
								000		001								
								000		001								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
1	EAG	000	MPY**	000	010					000		000						
			DIV**	000	100							000		000				
			DLD**	100	010							000		000				
			DST**	100	100							000		000				
			ASR	001	000													
			ASL	000	000													
			LSR	001	000													
			LSL	000	000													
			RRR	001	001													
			RRL	000	001													
														0	11			
														0	00			
														1	0			
										1	0							
										0	0							
										0								
NOTES:										†CLE		Only this bit is required						
D/I, A/B, Z/C, D/E, H/C coded 0/1										‡SL		Only this bit and bit 11 (A/B as applicable) are required						
** Second word is Memory Address																		

Extended Instruction Group Codes in Binary (continued)

Memory Expansion

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
DJP/DJS	1	0	0	0	1	0	1	1	1	1	0	1	1					
														DJP	=	0	1	0
														DJS	=	0	1	1
SYB/USB/PAB/ PBB/SSM/JRS	1	0	0	0	1	0	1	1	1	1	0	0	1					
														SYB	=	0	0	0
														USB	=	0	0	1
														PAB	=	0	1	0
														PBB	=	0	1	1
														SSM	=	1	0	0
														JRS	=	1	0	1
XMA/XLA/XSA/ XCA/LFA	1	0	0	0	0	0	1	1	1	1	0	1	0					
														XMA	=	0	1	0
														XLA	=	1	0	0
														XSA	=	1	0	1
														XCA	=	1	1	0
														LFA	=	1	1	1
MB/MBF/MBW/ MWI/ MWF/MWW	1	0	0	0	1	0	1	1	1	1	0	0	0					
														MBI	=	0	1	0
														MBF	=	0	1	1
														MBW	=	1	0	0
														MWI	=	1	0	1
														MWF	=	1	1	0
														MWW	=	1	1	1
SYA/USA/ PAA/PBA	1	0	0	0	0	0	1	1	1	1	0	0	1					
														SYA	=	0	0	0
														USA	=	0	0	1
														PAA	=	0	1	0
														PBA	=	0	1	1

Extended Instruction Group Codes in Binary (continued)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XMM/XMS/ XMB/XLB/ XSB/XCB/LFB	1	0	0	0	1	0	1	1	1	1	0	1	0			

XMM = 0 0 0
 XMS = 0 0 1
 XMB = 0 1 0
 XLB = 1 0 0 0
 XSB = 1 0 1
 XCB = 1 1 0
 LFB = 1 1 1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSA/RVA	1	0	0	0	0	0	1	1	1	1	0	1	1			

RSA = 0 0 0
 RVA = 0 0 1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSB/RCB/SJP/ SJS/UJP/UJS	1	0	0	0	1	0	1	1	1	1	0	1	1			

RSB = 0 0 0
 RVB = 0 0 1
 SJP = 1 0 0
 SJS = 1 0 1
 UJP = 1 1 0
 UJS = 1 1 1

System Communications Area Locations

Octal Location	Contents	Description
System Table Definition		
01645	XIDEX	Address of current program's ID extension
01646	XMATA	Address of current program's MAT entry
01647	XI	Address of index register save area
01650	EQTA	FWA of Equipment Table
01651	EQT#	Number of EQT entries
01652	DRT	FWA of Device Reference Table, word 1
01653	LUMAX	Number of logical units in DRT
01654	INTBA	FWA Interrupt Table
01655	INTLG	FWA of Interrupt Table Entries
01656	TAT	FWA of Track Assignment Table
01657	KEYWD	FWA of Keyword Block
I/O Module/Driver Communion		
01660	EQT1	} Addresses of first 11 words of current EQT entry (see 01771 for last four words)
01661	EQT2	
01662	EQT3	
01663	EQT4	
01664	EQT5	
01665	EQT6	
01666	EQT7	
01667	EQT8	
01670	EQT9	
01671	EQT10	
01672	EQT11	
01673	CHAN	} Current DCPD channel number I/O address of time-base card EQT entry address of system TTY
01674	TBG	
01675	SYSTY	
System Request Processor/Exec Communion		
01676	RQCNT	} Number of request parameters -1 Return point address
01677	RQRTN	
01700	RQP1	} Addresses of request parameters (set for a maximum of nine parameters)
01701	RQP2	
01702	RQP3	
01703	RQP4	
01704	RQP5	
01705	RQP6	
01706	RQP7	
01707	RQP8	
01710	RQP9	
System List Addresses		
01711	SKEDD	Schedule list
01712	PVCN	Privileged nest counter
01713	SUSP2	Wait Suspend list
01714	SUSP3	Available Memory list
01715	SUSP4	Disc Allocation list
01716	SUSP5	Operator Suspend list

System Communications Area Locations (continued)

Octal Location	Contents	Description
Program ID Segment Definition		
01717	XEQT	ID segment address of current program
01721	SLINK	Linkage
01721	XTEMP	Temporary (five words)
01726	XPRI0	Priority word
01727	XPENT	Primary entry point
01730	XSUSP	Point of suspension
01731	XA	A-Register at suspension
01732	XB	B-Register at suspension
01733	XEO	E and overflow register suspension
System Module Communion Flags		
01734	OPATN	Operator/keyboard attention
01735	OPFLG	Operator communication flag
01736	SWAP	RT disc-resident swapping flag
01737	DUMMY	I/O address of dummy interface flag
01740	IDSDA	Reserved
01741	IDSDP	Reserved
Memory Allocation Bases Definition		
01742	BPA1	FWA user base page link area
01743	BPA2	LWA user base page link area
01744	BPA3	FWA user base page link
01745	LBORG	FWA of resident library area
01746	RTORG	FWA of real-time COMMON
01747	RTCOM	Length of real-time COMMON
01750 D	RTDRA	FWA of real-time partition
01751 D	AVMEM	LWA - 1 of real-time partition
01752	BGORG	FWA of background COMMON
01753	BGCOM	Length of background COMMON
01754 D	BGDRA	FWA of background partition
Utility Parameters		
01755	TATLG	Negative length of track assignment table
01756	TATSD	Number of tracks on system disc
01757	SECT2	Number of sectors/track on LU 2 (system)
01760	SECT3	Number of sectors/track on LU 3 (aux.)
01761	DSCLB	Disc address of library entry points
01762	DSCLN	Number of user available library entry points
01763	SDCUT	Disc address of relocatable disc-resident library
01764	SYSLN	Number of system library entry points
01765	LGOTK	LGO: LU#, starting track, number of tracks (same format at ID segment word 26) Current LGO track/sector address (same format at ID segment word 26)
01766	LGOC	LS: LU# and disc address (same format at ID segment word 26) Memory protect ON/OFF flag (0/1)
01767	SFCUN	
01770	MPTFL	} Address of last four words of current EQT
01771	EQT12	
01772	EQT13	
01773	EQT14	
01774	EQT15	
01775 D	FENCE	Memory protect fence address
01776	VMASWP	VMA swap flag
01777	BGLWA	LWA memory background partition
The letter D indicates the contents of the location are set dynamically by the dispatcher.		

Device Reference Table (DRT)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved		Subchannel #						EQT Entry Number								word 1
F	Downed I/O Request List Pointer															word 2
Odd LU Lock								Even LU Lock								byte 5

where:

F (up/down flag) 0 if device is up
 1 if device is down

LU Lock 0 if no lock on the LU, otherwise
 the resource number being used for the lock

NOTE

A cartridge lock on an FMP cartridge does not affect the LU lock byte.

Equipment Table (EQT)

Word	Contents															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	R	I/O Request List Pointer <C>														
2	R	Driver Initiation Section Address <A>														
3	R	Driver Continuation/Completion Section Address <A>														
4	D <A>	B 	P <E>	S <E>	T <C>	Lower 5 bits of Subchannel <C>					I/O Select Code # <A>					
5	AV <F>		EQUIPMENT TYPE CODE <A>						STATUS <E>							
6	CONWD (Current I/O Request Word) <C>													MSB <C>		
7	Request Buffer Address <C>															
8	Request Buffer Length <C>															
9	Temporary Storage <D> or Optional Parameter <C>															
10	Temporary Storage <D> or Optional Parameter <C>															
11	Temporary Storage for Driver <D>															
12	Temporary Storage for Driver <D>							or	EQT Extension Size, if any <A>							
13	Temporary Storage for Driver <D>							or	EQT Extension Starting Address, if any <A>							
14	Device Timeout Reset Value 															
15	Device Timeout Clock <C>															

Legend for EQT Table

R = reserved for system use.	
I/O Request List Pointer	= points to list of requests queued up on this EQT entry
D	= 1 if DCPC required
B	= 1 if automatic output buffering used.
P	= 1 if driver is to process power fail.
S	= 1 if driver is to process timeout.
T	= 1 if device timed out (system sets to zero before each I/O request)
Subchannel	= last subchannel addressed. (lower 5 bits)
MSB	= most significant bit of the subchannel (bit 6).
I/O Select Code	= I/O select code for the I/O controller (lower number if a multi-board interface).
AV	= I/O controller availability indicator: 0 = available for use. 1 = disable (down) 2 = busy (current in operation). 3 = waiting for an available DCPC channel.
Equipment Type Code	= type of device on this controller. When this octal number is linked with "DVy," it identifies the device's software driver routine. Some standard driver numbers are:
	00 to 07 = paper tape devices or consoles 00 = teleprinter or keyboard control device 01 = photoreader 02 = paper tape punch 05 = 264x-series terminals 07 = multi-point devices
	10 to 17 = unit record devices 10 = plotter 12 = line printer 15 = mark sense card reader
	20 to 37 = magnetic tape/mass storage devices 23 = 9-track magnetic tape (800/1600 BPI) 31 = 7900 moving head disk 32 = 7905/06/20/25 moving head disk drive, cartridge tape drive or 9895 flexible disk drive. 36 = writable control store 37 = HP-IB
	40 to 77 = instruments
STATUS	= actual physical status or simulated status at the end of each operation (see Device Status Table).
CONWD	= combination of user control work and user request code word in the I/O EXEC call (see EQT wd. 6).
Letters in brackets (<>) indicate the nature of each data item as follows:	
<A>	= fixed at generation or reconfiguration time; never changes
	= fixed at generation or reconfiguration time; can be changed online
<C>	= set up or modified at each I/O initialization
<D>	= available as temporary storage by driver
<E>	= can be set by driver
<F>	= maintained by system

Device Status Table A

Device\Status	7	6	5	4	3	2	1	0
Teleprinter(s) Photoreader(s) Punch(s) DVR00	X	—	End of I/O Tape	—	—	STL	TEN	—
262x 263x 264x Terminal Cartridge Tape Unit DVR05, DVA05	BF EOF	— TLP	CD EOT	— RE	— LCA	— CWP	TEN EOD	— CNI/ DP
2892A Card Reader DVR11	HE/ SOR	SF	HE/ SF	PF	TE/ PF	OL	ICC/ HRF	RNR
2607 Line Printer 2610 Line Printer 2613/17/18 Line Printer 2631 Line Printer DVA12	— — — —	TOF TOF TOF TOF	— — — —	ID ID ID BR	PSE SSE ON ON	OL PO NR PO	— — V9 —	— — V12 —
2608A Line Printer DVR12	PW	TOF	S8	VI	ON	NR	V9	V12
2767 Line Printer DVR12	TOF	DM	ON	RX	—	—	APE	—
7261A Card Reader DVR15	EOF	—	HF/ SF	PF	—	—	DE	RNR
7970 Mag Tape DVR23, DVS23	EOF	ST	EOT	TE	I/O R	NW	PE/ TE	OL
7900 Moving Head Disk DVR31	—	NR	EOT	AE	FC	SC	DE	EE
79XX Disk Drives DVR32	PS	FS	HF	FC	SC	NR	DB	EE
79XXH,9895 Disk Drives DVA32	PS	FS	HF	FC	SC	NR	DB	EE
CS/80 Disk Drives DVM33 See Status Table B DVR33	WP	REF/ UD	EOF/ EOV	UN	FA	NR	CHE	SE
59310B HP-IB DVR37	—	EF	I/O	NOA	SRQA	IFC	TO	—
Terminal Cartridge Tape Unit DV800	EOF EOF	BRK TLP	EOT EOT	LD RE	OF LCA	PFE CWP	TO EOD	Error CNI/ DP
DVC00	—	—	EOT	—	—	—	TO	Error
2563A 2564A 2565 2566 2608S DVC12	PW	—	—	—	ON	—	—	—
DVM00	—	BRK	EOT	LD	PE/OV	TAD	TEN	TO

Device Status Table B

DVR33 127323A,127733A Disk Drives	
Bits 0–7	Meaning
00000000	No Error
00000011	No Drive Power
00000101	Door Open
00000111	No Disk
00001011	Record Not Found
00001101	Track Not Found
00001111	Data Checkword Error
00010001	Data Overrun
00010011	Read "Tight Margin" Error
00011111	Transfer Incomplete
00100001	Data Block Too Long
00100000*	End of Track (Access Track > 66)
01000000*	Disk Change
10000000*	Disk Write Protected
DVA47 Serial Link Drive	
Bits 0–7	Meaning
00000001	Timeout Occurred
00000010	Hardware Failure
00000011	Hardware Failure on Controller
00000100	Bad System Configuration
00000101	Illegal Request

Device Status Table Key

AE = Address Error	NW = No write
AF = Abort Flag (NR (Bit = 7 = 0) has occurred during/since last data transfer)	(ring missing or rewinding)
APE = Auto Page Eject	OF = FIFO Overflow
BF = Buffer Flushed	OL = Offline
BR = Buffer Ready	ON = Online
BRK = Break Detected	OV = Overflow Detected
BT = Broken Tape	PD = Pen Down
CD = Control-D Entered	PE = Parity Error
CE = Compare Error	PF = Pick Fail
CHE = Channel Error	PFE = Parity or Framing Error
CNI = Cartridge Not Inserted	PW = Powerfail
CWP = Cartridge Write Protected	PO = Paper Out
DB = Device Busy	PS = Protect Switch Set
DE = Data Error	PSE = Print Switch Enabled
DF = Drive or Controller Fault	RD = Release Drive
DM = Demand (1=idle)	RE = Read Error
DR = Disk Ready	RER = Recoverable Error
EE = Error Exists	RNR = Reader Not Ready
EF = EQT Extension Area Full	RX = Ready (0= Poweron)
EOD = End of Data	SAC = Sector Address Coincidence
EOF = End of File	SC = Seek Check
EOT = End of Track/TAPE	SE = Severe Error
EOV = End of Volume	SF = Stacker Full
Error = Severe Error Occurred	SOR = EOF Switch On During Read
FA = Drive or Controller Fault	SSE = Start Switch Enabled
FC = Flagged Track	ST = Start of Tape
FS = Driver Format Switch is Set	STL = Stall Required In Program
HE = Hopper Empty	S8 = Set is 8 LPI
HF = Hardware Fault	TAD = Type-ahead data available
ICC = Illegal Card Code	TE = Timing Error
ID = Idle	TEN = Terminal
IFC = IFC Detected	TLP = Tape at Load Pt
I/O = Illegal I/O Request	TO = Device Timeout
I/OR = I/O Reject	TOF = Top of Form
LCA = Last Command Aborted	UD = Unrecoverable Data
LCF = Last Character Flag	UN = Uninitialized Media
LD = Line Down	VI = VFC Initialized
NE = No Error	V9 = VFU Chan 9 Detected
NOA = Non-existent alarm program	V12 = VFU Chan 12 Detected
NR = Not Ready	WE = Currently addressed track is Write Enabled
	WP = Write Protect
	X = Driver internal use

EQT Word 6



Other subfunctions are driver specific and may or may not be defined.

ID Segment

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Word
List Linkage															0	← XEQT
TEMP 1															1	}
TEMP 2															2	
TEMP 3															3	
TEMP 4															4	
TEMP 5															5	
Priority Primary Entry Point															6	} Memory-Resident Programs
Point of Suspension A-Register															7*	
B-Register															8	
EO-Registers															9	
Name 1															10	
Name 2															11	
Name 3															12*	
Name 4															13*	
Name 5															14*	
TM	ML	TS	SS	Type											15	
NA	NS	NP	W	A	FS	O	LP	R	D	/	/	Status			16	
Time List Linkage															17	
RES		T		Multiple											18	
Low Order 16 Bits of Time															19	
High Order 6 Bits of Time															20	
BA	FW	M	AT	RM	RE	PW	RN	Father ID Segment No.							21	
RP	# pgs. (no BP)				MPFI			DE	Partition No. - 1						22*	
Low Main Address															23*	
High Main Address + 1															24*	
Low Base Pg Addr (Non-MLS Prog) or # Sectors on LU 2 or 3 for MLS Prog															25*	
High Base Page Address + 1															26*	
SD	Program: Track #														27	
LU	/	/	Debug			Swap: Track #									28	
ID Extension No.							EMA Size								29	
High Address + 1 of Largest Segment or Node															30	
Timeslice Word															31	
SEQCNT			SH	DC	CP	DS	Session ID								32	
SCB Pointer															33	
MS	# Pages Disc Resident					# Pages Memory-Resident									34	
MP	# Pgs Dyn.Buffer Area				E	/	# of Swap Tracks								35*	
Start Sector Address of Program								LU # of Prog							36*	

* Words used in short ID segments for program segments

ID Segment (continued)

TM	=	Temporary Load (copy of ID segment is not on the disk)
ML	=	Memory Lock (program may not be swapped)
TS	=	The program is Transportable
SS	=	Short Segment (indicates a 9-word segment)
TYPE	=	Specified program Type (1–6)
NA	=	No Abort (instead, pass abort errors to program)
NS	=	No Suspension on I/O requests (instead, pass control to program)
NP	=	No Parameters allowed on reschedule
W	=	Wait bit (waiting for program whose ID segment address is in word 1)
A	=	Abort on next list entry for this program
FS	=	File System bit
O	=	Operator suspend on next schedule attempt
LP	=	Load In Progress; program is being dispatched from disk
R	=	Resource save (save dormant on next schedule attempt)
D	=	Dormant bit (set control on next schedule attempt)
Status	=	Current program status
T	=	Time list entry bit (program is in the time list)
BA	=	Batch (program is running under batch)
FW	=	Parent Is Waiting (parent scheduled with wait)
M	=	Multi-Terminal Monitor bit
AT	=	Attention bit (operator has requested attention)
RM	=	Re-entrant Memory must be moved before dispatching program
RE	=	Re-entrant routine now has control
PW	=	Program Wait (another program wants to schedule this one)
RN	=	Resource Number either owned or locked by this program
RP	=	Reserved Partition (only for programs that request it)
MPFI	=	Memory Protect Fence Index
DE	=	Defer EXEC 6 (terminate program) request
SD	=	Clear this ID segment when the program terminates
LU	=	0 if LU 2, 1 if LU 3
Debug	=	Debug state; 0 = not, 1 = debugging, 2 = monitor, 3 = wait for initial loader or \$LIBX.
SEQCNT	=	Sequence Counter
SH	=	Shareable EMA flag (program or progeny uses shareable EMA)
DC	=	Don't Copy flag
CP	=	Copy Flag
DS	=	DS program
Session	=	System LU of terminal where program was loaded
ID		
MS	=	Multi-level segmentation flag
MP	=	Program is using Modified maps for I/O
E	=	EXEC 4 (track allocation) request was made by program

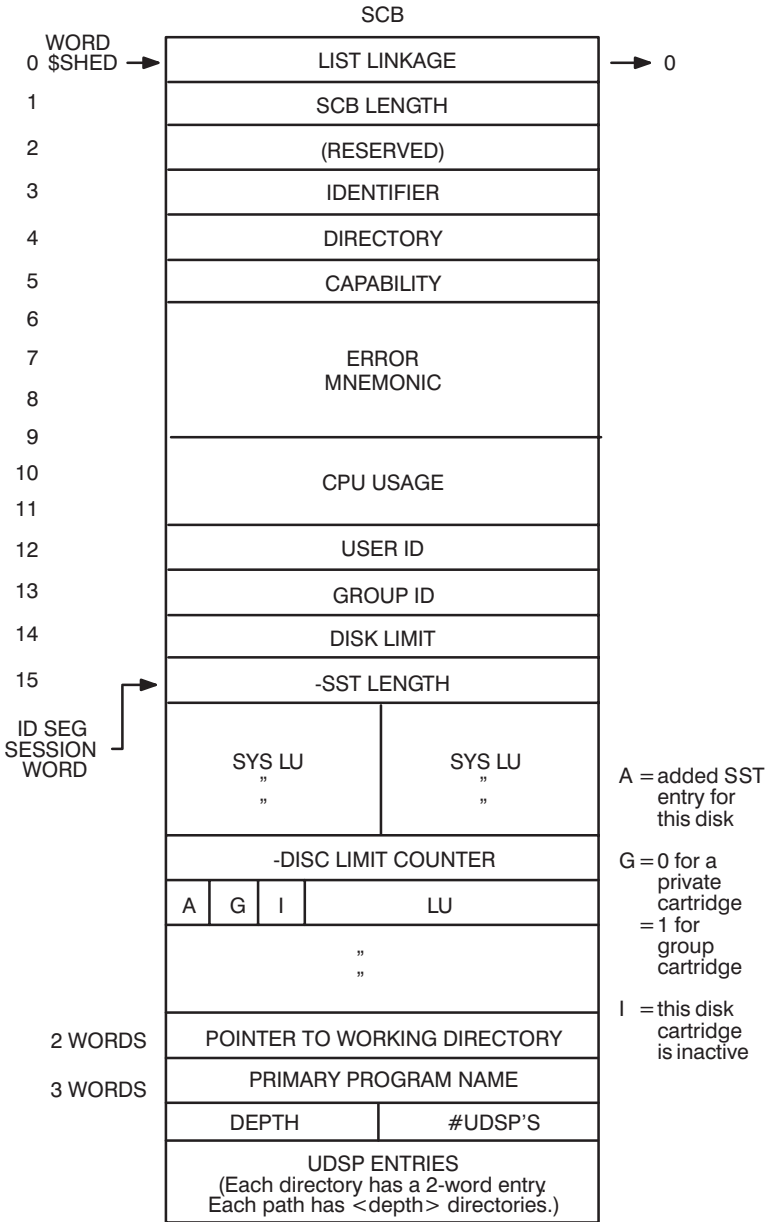
ID Segment Extension

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	Not Used										MSEG Size					Word 0
MSEG START PAGE (LOGIC)					DE	(PHYSICAL)EMA START PAGE										Word 1
S	/ / / / /		SW	/ / / / /			Index # Into \$EMTB									Word 2
Last Virtual Page #															Word 3	
Usage Count Minus 1 For SHEMA5 Otherwise Reserved															Word 4	

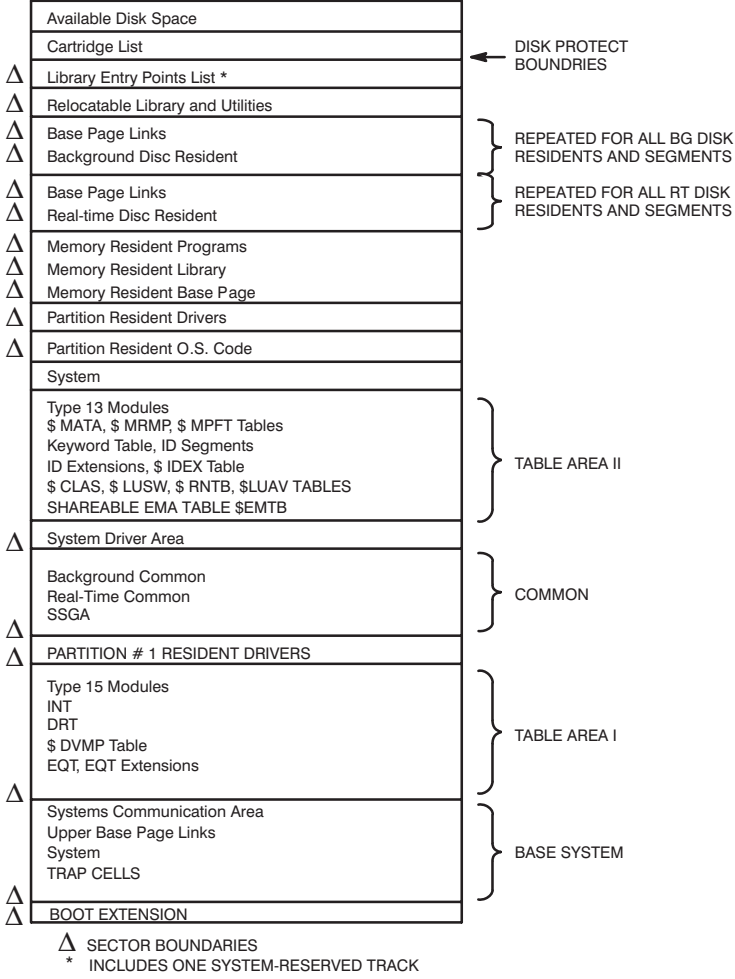
where:

- DE 0 if the EMA size was specified by the user
- DE 1 if the EMA size is allowed to default to the maximum size available to the system
- S 1 if the program is using shareable EMA
- SW 0 PTE table does not contain valid data
- SW 1 PTE table is still intact

Session Control Block (SCB)



System Disk Layout



Device File DCB

Word 0	0
1	0
2	File Type (0)
3	XLUEX LU Word
4	XLUEX Function Word
5	Spacing Flags
6	EOF Function Code
7	Read/Write Flags
8	0
9	Program ID Segment Address
10	0
11	0
12	0
13	32-bit Record Number
14	
15	0

Data Control Block (DCB) Format

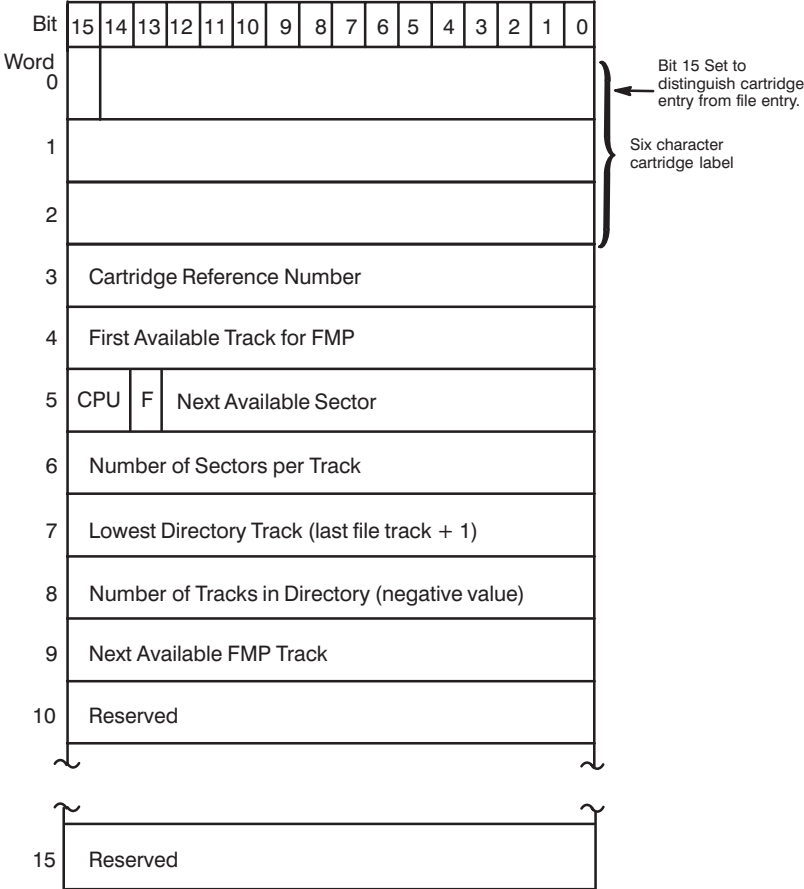
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Word 0	Sector Off-set			Block # of File Directory						DS	LU # of File Directory or of File if on Disk						} File Directory Address	
1	Track # of File Directory																	
2	File Type (May be Overrirdden at Open, Unless Type 0)																	
3	Track Address of File (Type ≥ 1)								OR	LU# of File (Type = 0)								
4	Sector Address of File (Type ≥ 1)								OR	EXEC Function Code (Type = 0)								
5	Fill Size in Chunks + sectors (Type ≥ 1)								OR	Spacing Code (Type = 0)								
6	Record Length (Type = 2)								OR	End-of-File Code (Type = 0)								
7	WA	PF	Number of Blocks in DCB Buffer						FM	EX	RA	OM	IB	EF	WR			
8	Number of Sectors per Track (Type ≥ 1)																	
9	Open/Close Indicator																	
10	Track # of Current File Position (Type ≥ 1)								OR	The Value of the A-Register after the Last EXEC Call (Type = 0)								} Current Position in File
11	Track # of Current File Position (Type ≥ 1)								OR	The Value of the B-Register after the Last EXEC Call (Type = 0)								
12	Location of Next Word in File (Type ≥ 1)																	
13	Record # of Current File																	
14	Position (Double Word Integer)																	
15	Extent Number (Type ≥ 3)																	
16	DCB Buffer Area (128 + N)																	

*File Directory Address

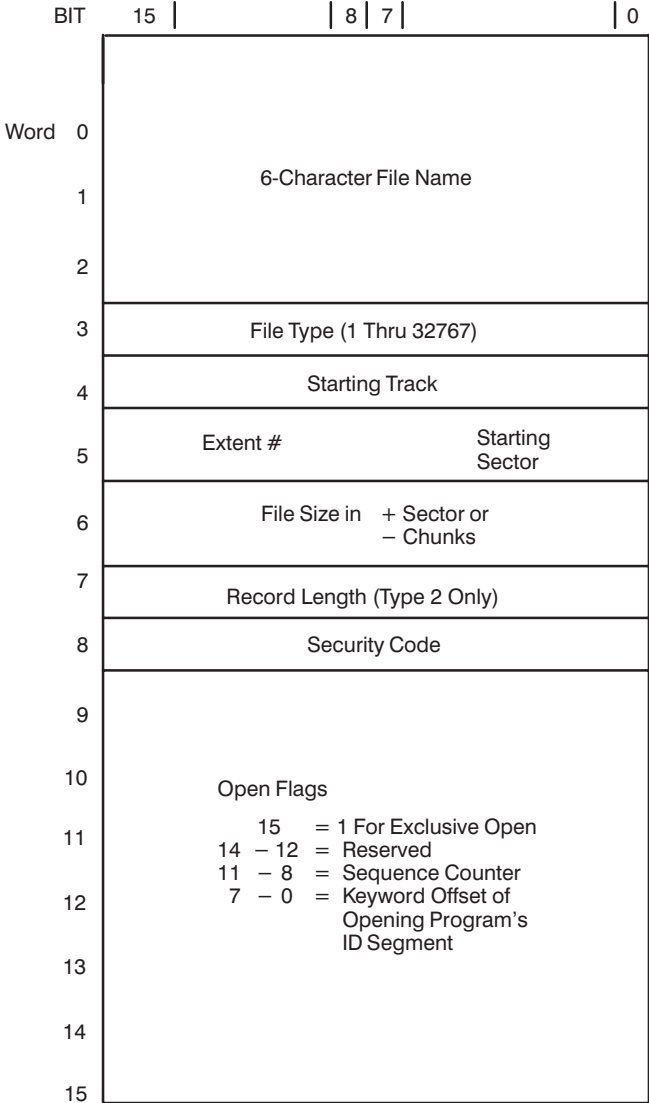
Legend For Data Control Block

Word	Content		
0	File Directory Address		
	Sector offset:	bits 15-13	= Entry offset from beginning of block (origin 0).
	Block offset:	bits 12-7	= Block offset from beginning of file directory track.
	(DS) Opened by DS:	bit 6	= 1 – Opened by DS transparency software.
	LU Number:	bits 5-0	= LU number.
5	Spacing Code: (type 0 file)	bits 15 bit 0	= 1 – Backspace legal. = 1 – Forward space legal.
6	End-of-File Code: (type 0 file)	01 10 11	lu = EOF on Magnetic Tape. lu = EOF on Paper Tape. lu = EOF on Line Printer.
7	Status Information		
	(WA) Write Allowed:	bit 15	= 1 – Write to file allowed = 0 – Write to file now allowed
	(PF) Partially Full:	bit 14	= 1 – DCB is only partially full = 0 – DCB is full
	DCB Buffer:	bits 13-7	= Number of blocks in the DCB buffer
	(FM) File Modify:	bit 6	= 1 – File has been modified = 0 – File has not been modified
	(EX) Extendable:	bit 5	= 1 – File is not extendable = 0 – File is extendable
	(RA) Read Allowed:	bit 4	= 1 – Read from file allowed = 0 – Read from file not allowed
	(OM) Open Mode:	bit 3	= 1 – Update open = 0 – Standard open
	(IB) In Buffer Flag:	bit 2	= 1 – Data in DCB buffer = 0 – Data not in DCB buffer
	(EF) EOF Read Flag:	bit 1	= 1 – EOF has been read = 0 – EOF has not been read
	(WR) To Be Written:	bit 0	= 1 – Data in DCB buffer to be written = 0 – Data in DCB buffer not to be written

FMGR Cartridge Specification Entry



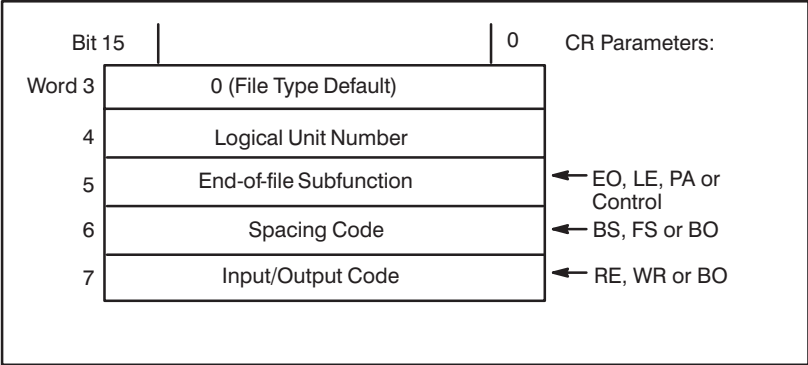
FMGR Directory File Entry



Word 0 = 0 If the Last Entry in Directory; = -1 if File is Purged

FMGR Type 0 File Directory Entry

The entries for non-disk (type 0) differ from those for disk files in words 3 through 7:



Word 5-7 are octal codes:

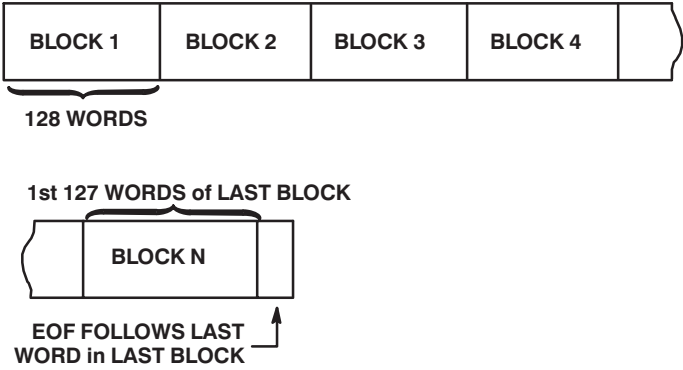
- end-of-file subfunction = 01LU for MT(EO)
 10LU for paper tape (LE)
 11LU for line printer (PA)
 or subfunction code

- spacing coded = bit 15 = 1 backspace legal (BS)
 bit 0 = 1 forward space legal (FS)

- input/output code = bit 15 = 1 input legal (RE)
 bit 0 = 1 output legal (WR)

Disk File Record Formats

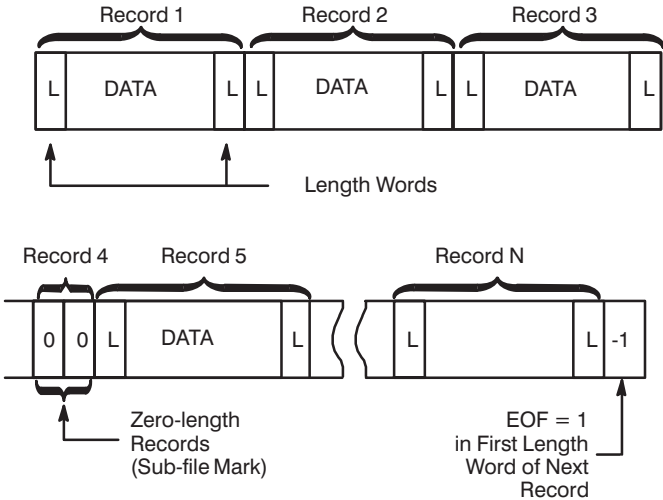
Fixed Length Formats (Types 1 and 2)



Type 1 Record length = Block length = 128 words

Type 2 Record length is user defined; may cross block boundaries but not past EOF.

Variable Length Formats (Types 3 and Above)



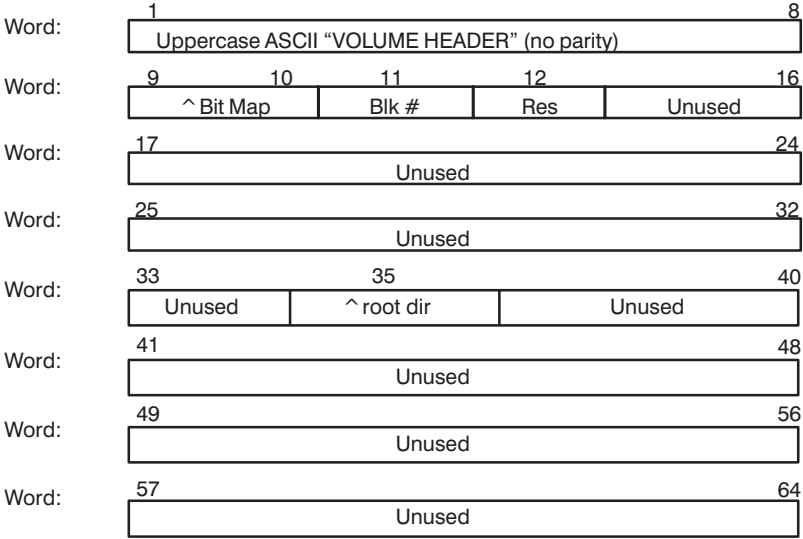
Type 6 File Format

Files created by the SP command or link as memory-image program files are always accessed by type 1 files (fixed length, 128-words per record).

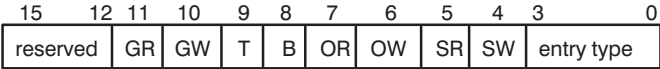
Word	Content	
0	-1	EOF unless forced to type 1
1-5	Not Used	
6	Priority	
7	Primary Entry Point	
8-11	Not Used	
12-13	Original Program Name	
14	Program Type	
15-16	Not Used	
17-19	Time Parameters	
20	Substatus 1 - Word 20 of ID Segment	
21	Substatus 2 - Word 20 of ID Segment	Words 0-35 and 38-42 contain program's ID-segment information
22	Low Main Address	
23	High Main Address + 1	
24	Low Base-page Address	
25	High Base-page Address + 1	
26	Program Track	
27	Swap Track	
28	ID Ext #/EMA Size	
29	High Address + 1 of Largest Segment	
30	Not Used	
31	Open Flag Word	Sum of contents of words 1650 thru 1657 and words 1742 thru 1747 and 1755 thru 1764 in base page.
32	Not Used	
33	Mix Word 1	
34	Mix Word 2	
35	Sector/LU of Program	
36	Checksum of Words 0 - 32	
37	Setup Code Word	
38	ID Extension - Word 0	
39	ID Extension - Word 1	
40	ID Extension - Word 2	
41	ID Extension - Word 3	If sign bit set, program file protected to this user ID.
42	ID Extension - Word 4	
43-45	Shared EMA Name	
46	Owner ID	If sign bit set, program file protected to this group ID
47	Owner's Group ID	
48	Capability Level Required	
49-96	Not Used	Minimum capability required to RU or RP this program.
97-104	Link's time stamp	
105-112	SNAP file used for Link.	
113-123	Time Type-6 File Created	
124-127	Not Used	

Remainder of file is an exact copy of the program being saved.

CI Volume Header Format

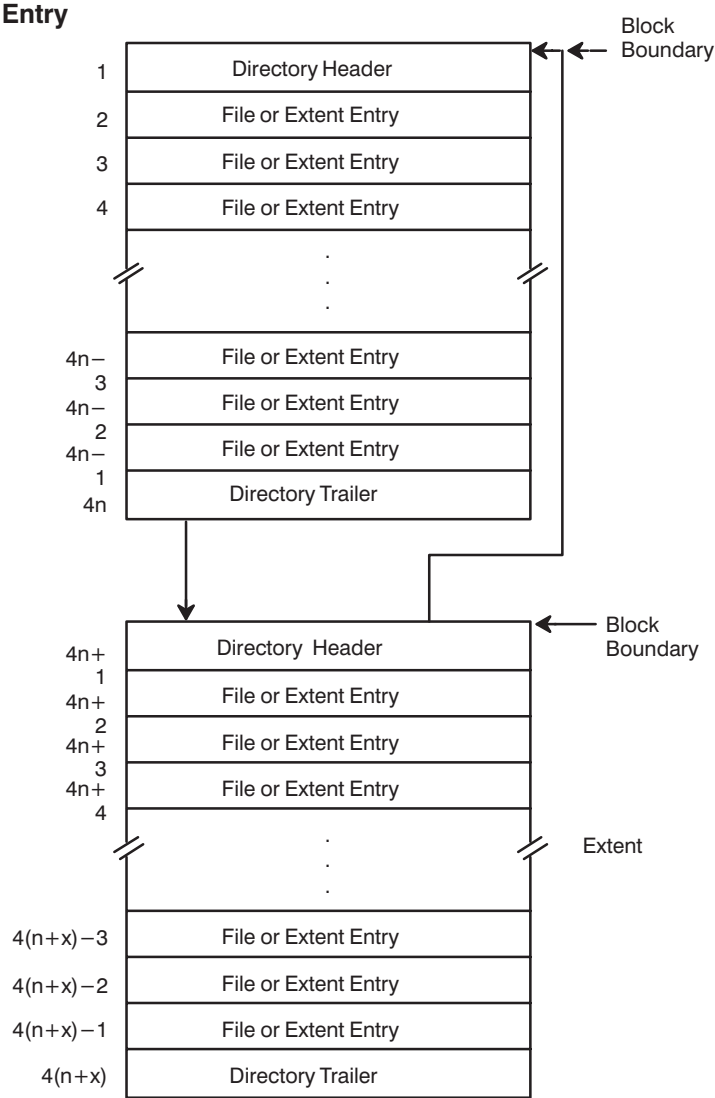


CI Directory Flag Word Format

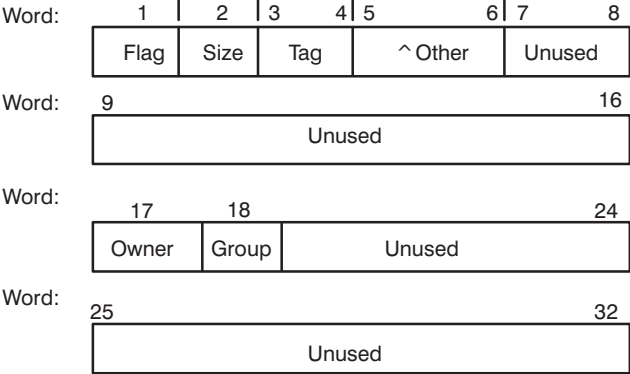


- † GR = group read access
 - † GW = group write access
 - †† T = temporary file
 - †† B = backup flag
 - † OR = owner read access
 - † OW = owner write access
 - † SR = system (others) read access
 - † SW = system (others) write access
 - entry type = 0 – empty, no files follow
 - 1 – directory header
 - 2 – directory trailer
 - 3 – file entry
 - 4 – extent entry
 - 5 – purged file entry
 - 6 – purged extent entry
 - 7 – empty, files may follow
 - 8 – 15 – reserved
- † Only used in a file entry and first header of a directory.
 †† Only used in a file entry.

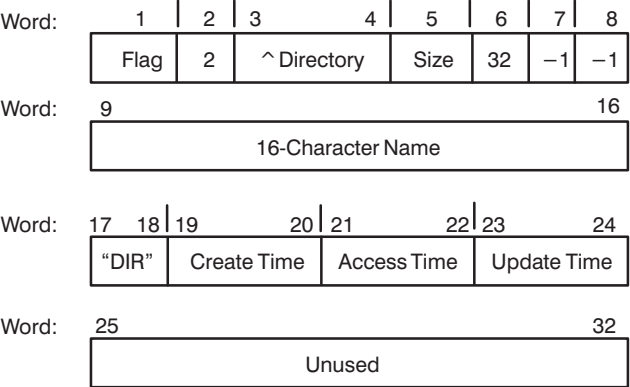
CI Directory Structure



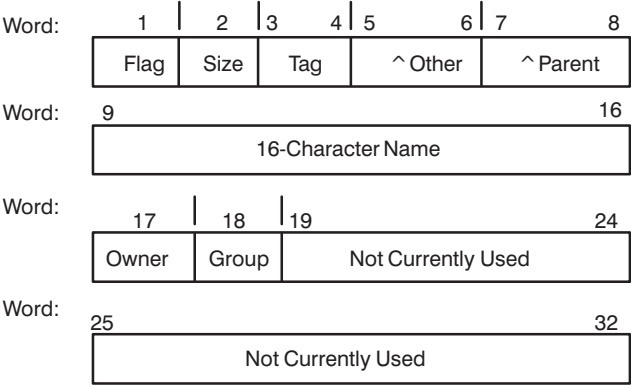
CI Root Directory Header/Trailer



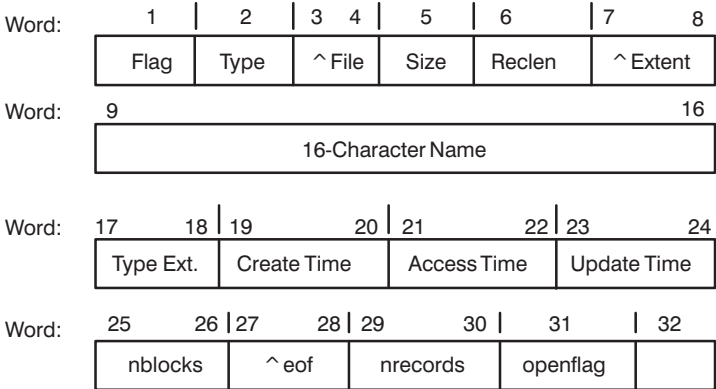
CI Root Directory Entry



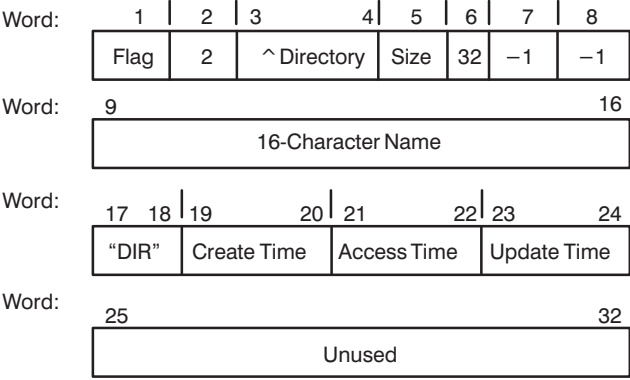
CI Directory Header/Trailer Format



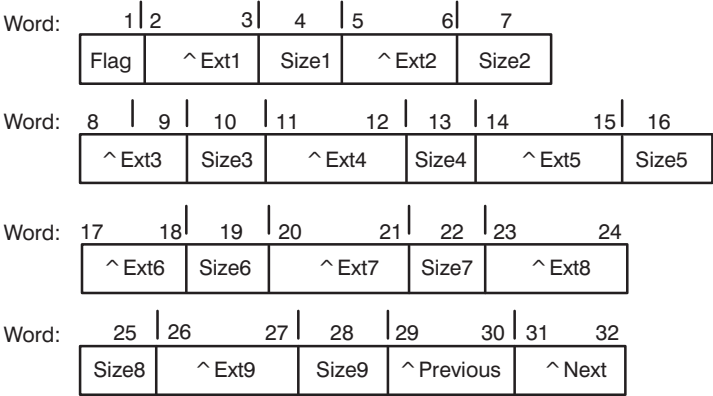
CI File Entry



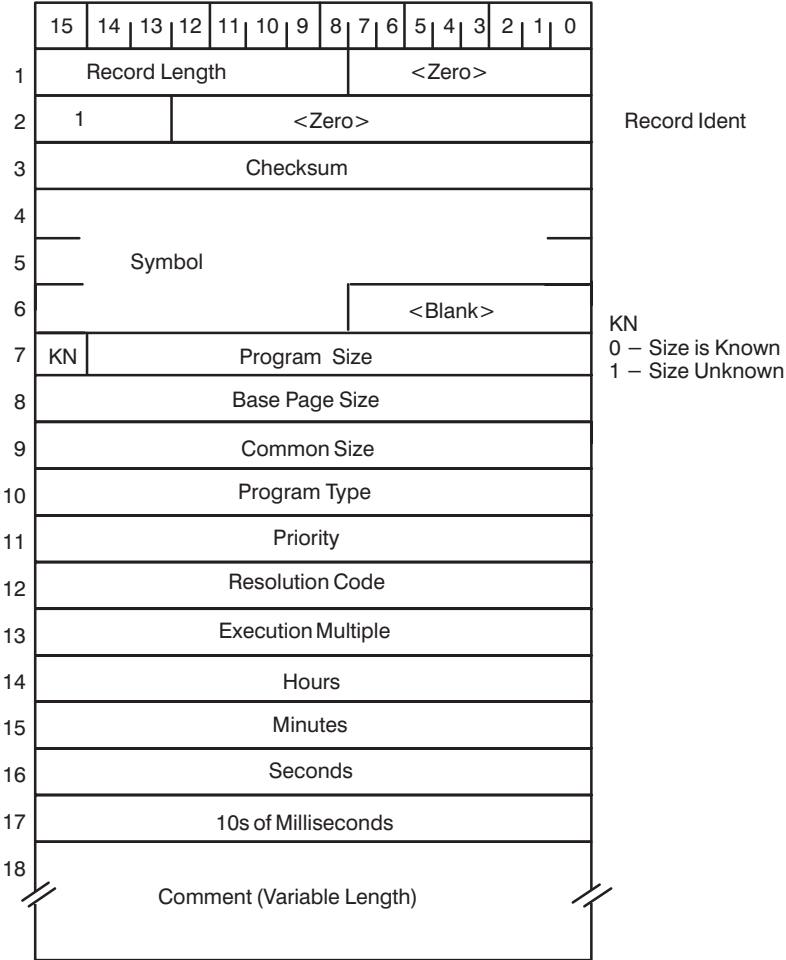
CI Subdirectory Entry



CI Extent Entry



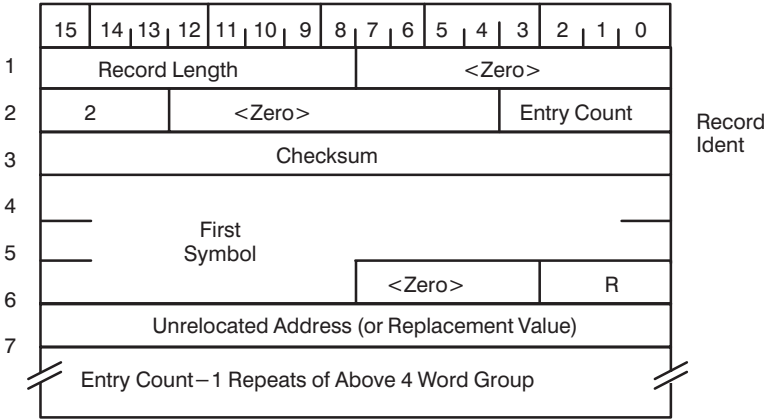
NAM Record



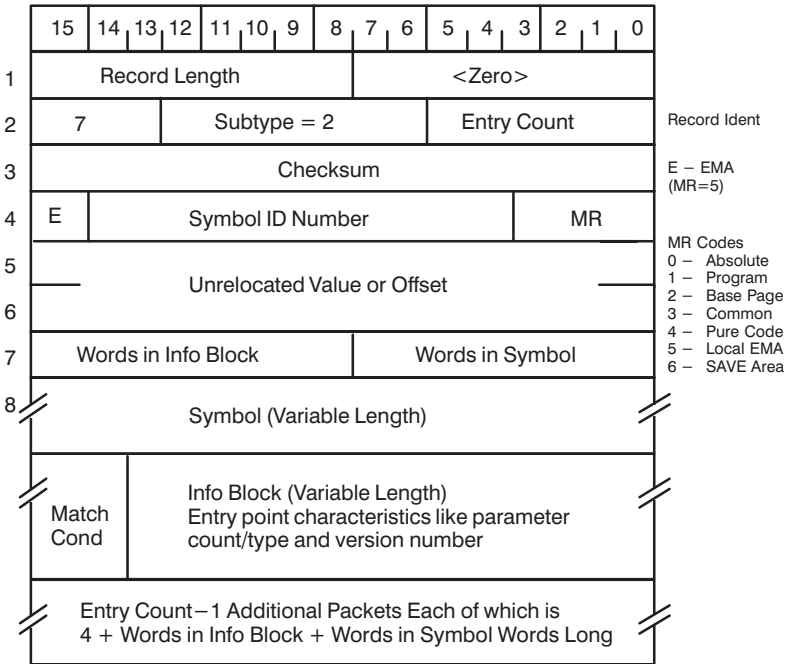
Extended NAM Record

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	Record Length								<Zero>								
2	7				Subtype = 1				Offset of Siz Wrđ				Record Indent				
3	Checksum																
4	Local EMA Size																
5																	
6	Save Size																
7	KN	Program Size															KN: 0 – Size is Known 1 – Size Unknown
8	Base Page Size																
9	Common Size																
10	Program Type																
11	Priority																
12	Resolution Code																
13	Execution Multiple																
14	Hours																
15	Minutes																
16	Seconds																
17	10s of Milliseconds																
18	C	Pure Code Size															C: 0 – Not CDS 1 – Cds Module
19	Year of Compilation																
20	Julian Day								Hour								
21	Minutes								Seconds								
22	Literal Count																
23	Revision of Producer (Days Since 1970)																
24	Revision of Preprocessor or –1 (Days Since 1970)																
25	Words of Comment								Words in Symbol								
26	Symbol (Variable Length)																
27	Comment (Variable Length)																
28	Length of File Name (–bytes)																
29	Source File Name (Variable Length)																

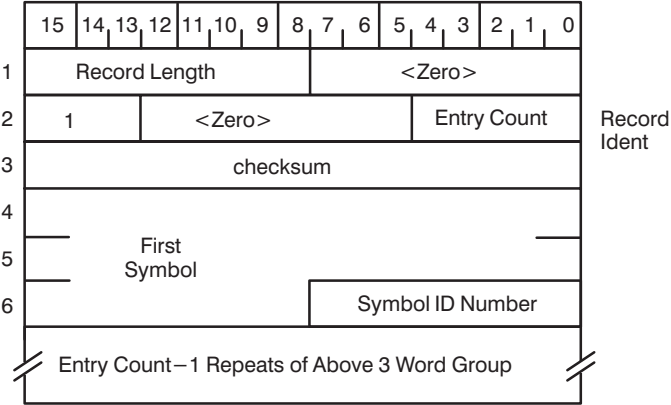
ENT Record



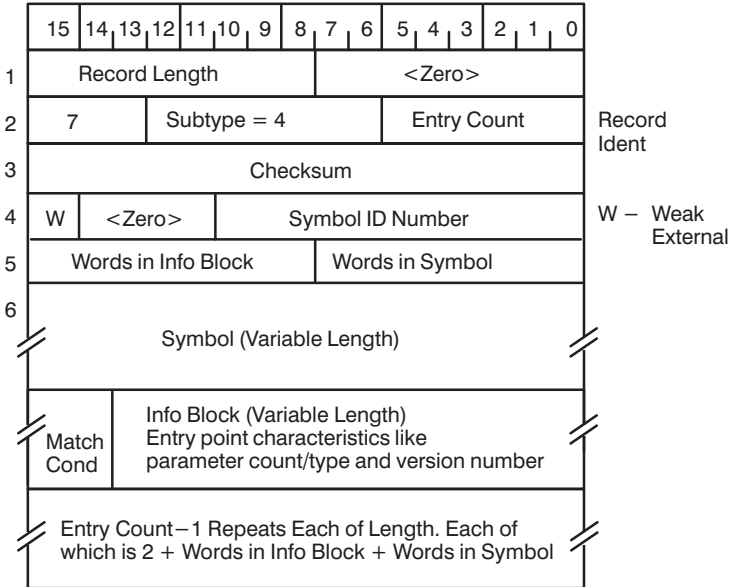
Extended ENT Record (XENT)



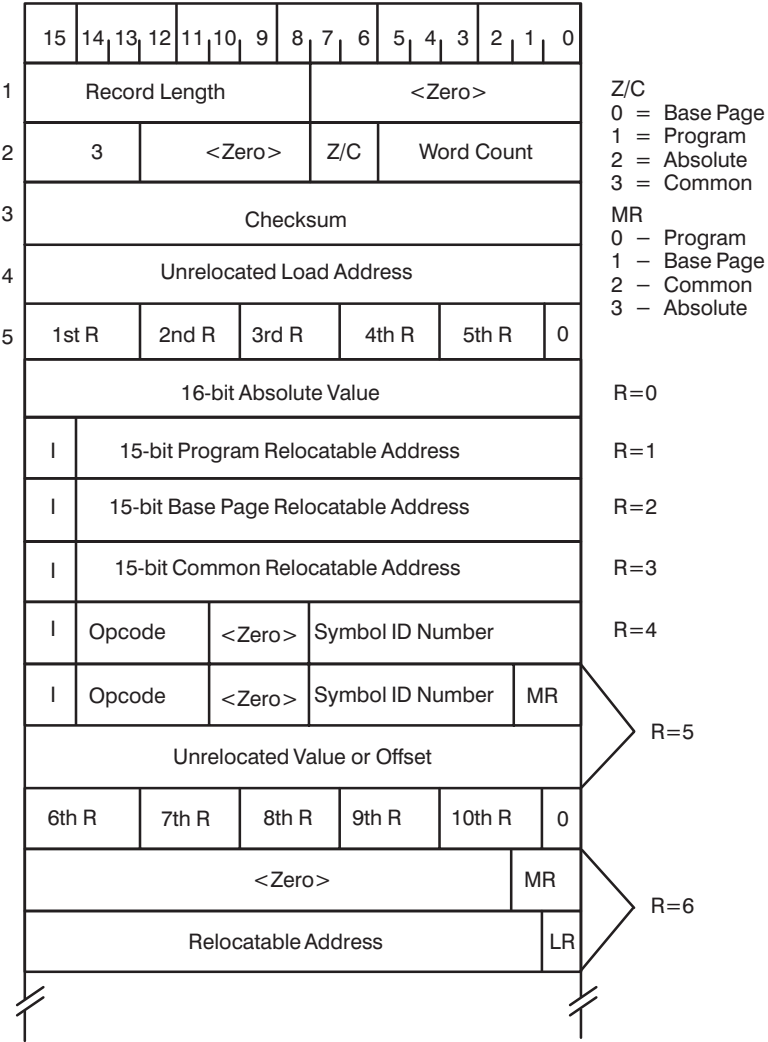
EXT Record



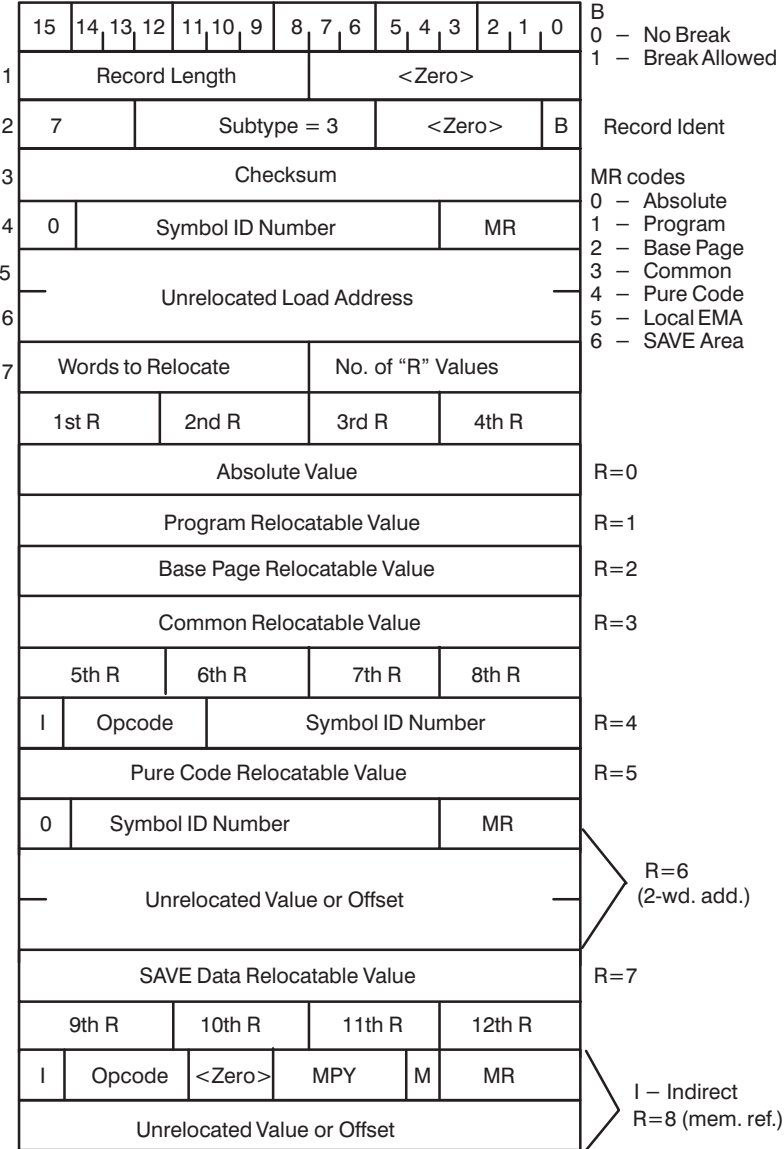
Extended EXT Record (XEXT)



DBL Record



Extended DBL Record (XDBL)



EMA Record

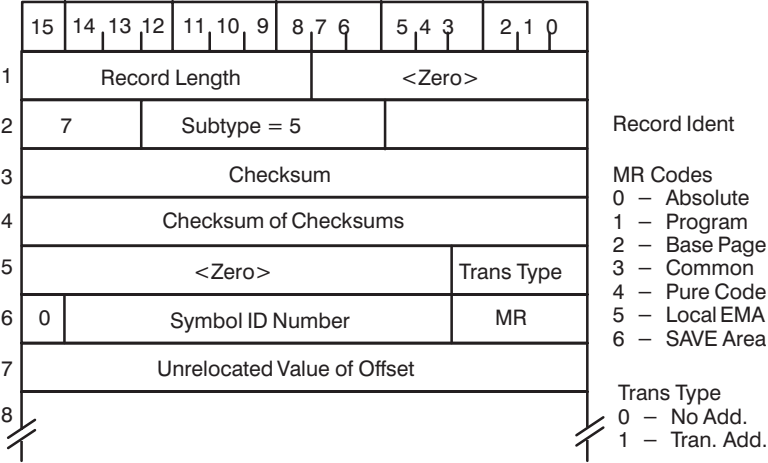
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Record Length								<Zero>							
2	6		<Zero>		EMA Size (in pages)											
3	Checksum															
4	Symbol															
5																
6												Symbol ID Number				
7	<Zero>											MSEG Size				

END Record

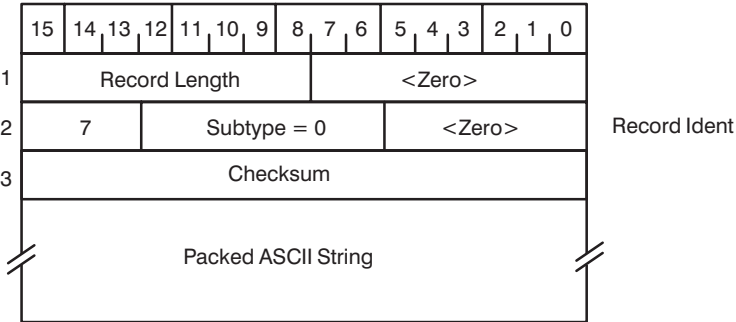
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Record Length								<Zero>							
2	5		<Zero>										R	T		
3	Checksum															
4	0	Unrelocated Transfer Address														

- T:
 0 – No Xfer. Add.
 1 – Xfer. Add.
- R:
 0 = Program
 1 = Base Page
 2 = Common
 3 = Absolute

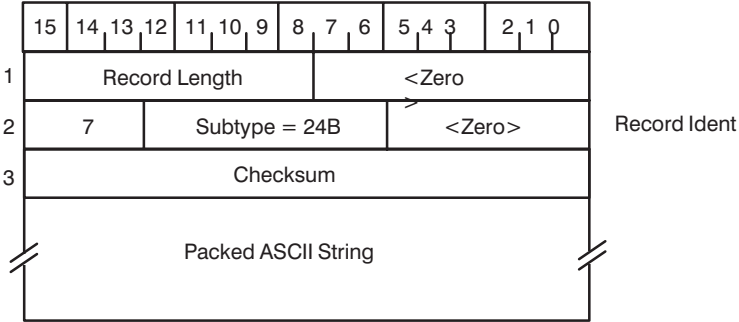
Extended END Record (XEND)



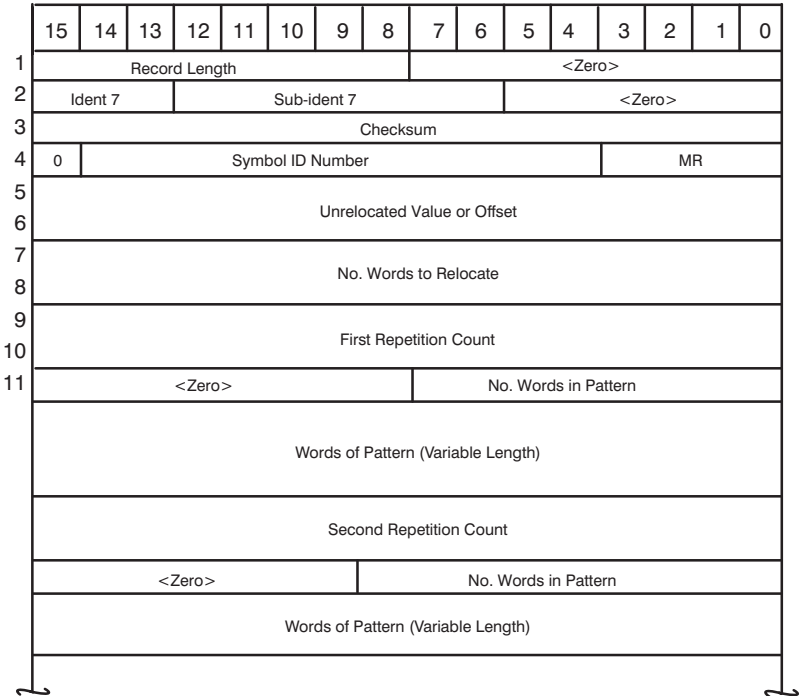
GEN Record



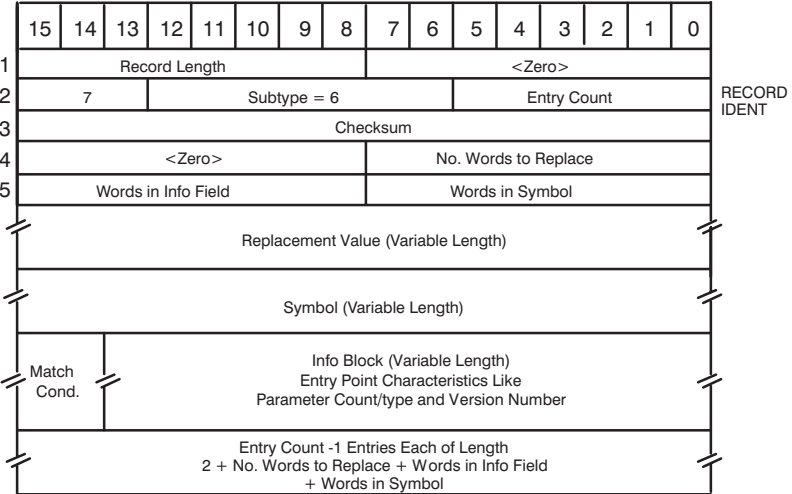
LOD Record



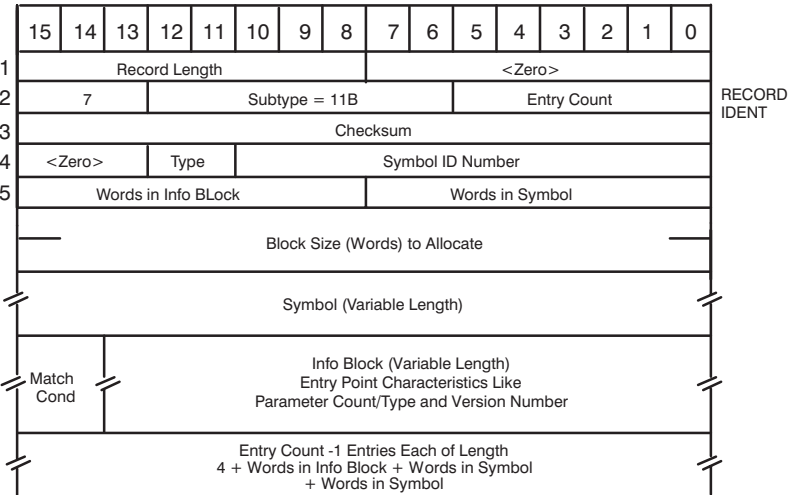
Data Record



RPL Record

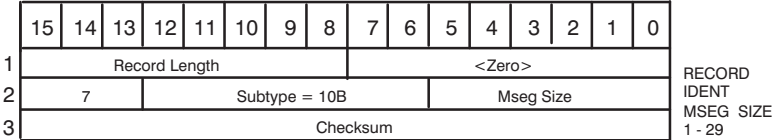


Allocate Record



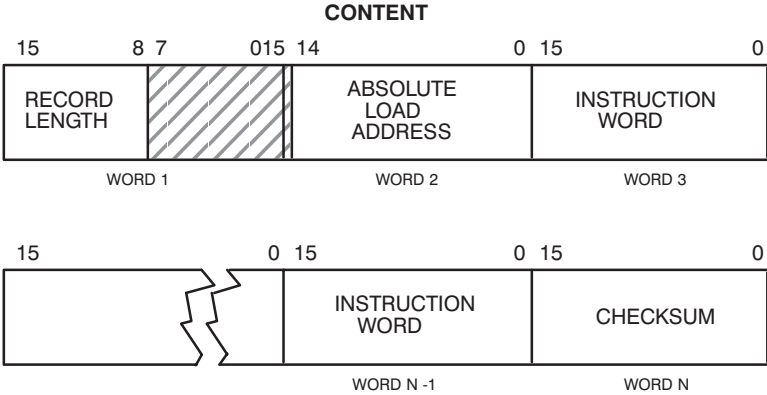
- Type = 0 Named COMMON (program allocate)
- Type = 1 Named SAVE COMMON (SAVE allocate)
- Type = 2 Named EMA COMMON (EMA allocate)
- Type = 3 Reserved

MSEG Record

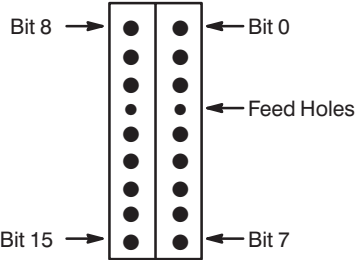


Absolute Tape Format

Absolute binary code is written to paper tape in the following format:



Each word represents two frames arranged as follows:



Explanation:

RECORD LENGTH = number of words in record excluding words 1 and 2 and the last word.

ABSOLUTE LOAD ADDRESS: starting address for loading the instructions which follow

INSTRUCTION WORDS: absolute instructions or data

CHECKSUM: arithmetic total of all words except first and last

FMGR Global Equivalence

S	G	P
0	-2	-48 Type
		-47 1
		-46 2
		-45 3
1	-1	-44 Type
		-43 1
		-42 2
		-41 3
2	-0	-40 Type
		-39 1
		-38 2
		-37 3
3	1	-36 Type
		-35 1
		-34 2
		-33 3
4	2	-32 Type
		-31 1
		-30 2
		-29 3
5	3	-28 Type
		-27 1
		-26 2
		-25 3
6	4	-24 Type
		-23 1
		-22 2
		-21 3
7	5	-20 Type
		-19 1
		-18 2
		-17 3
8	6	-16 Type
		-15 1
		-14 2
		-13 3
9	7	-12 Type
		-11 1
		-10 2
		-9 3
10	8	-8 Type
		-7 1
		-6 2
		-5 3
11	9	-4 Type
		-3 1
		-2 2
		-1 3
12	10	0 Type
		1 1
		2 2
		3 3
13	11	4 4
		5 5
		6 6
		7 7
		8 8
		9 9

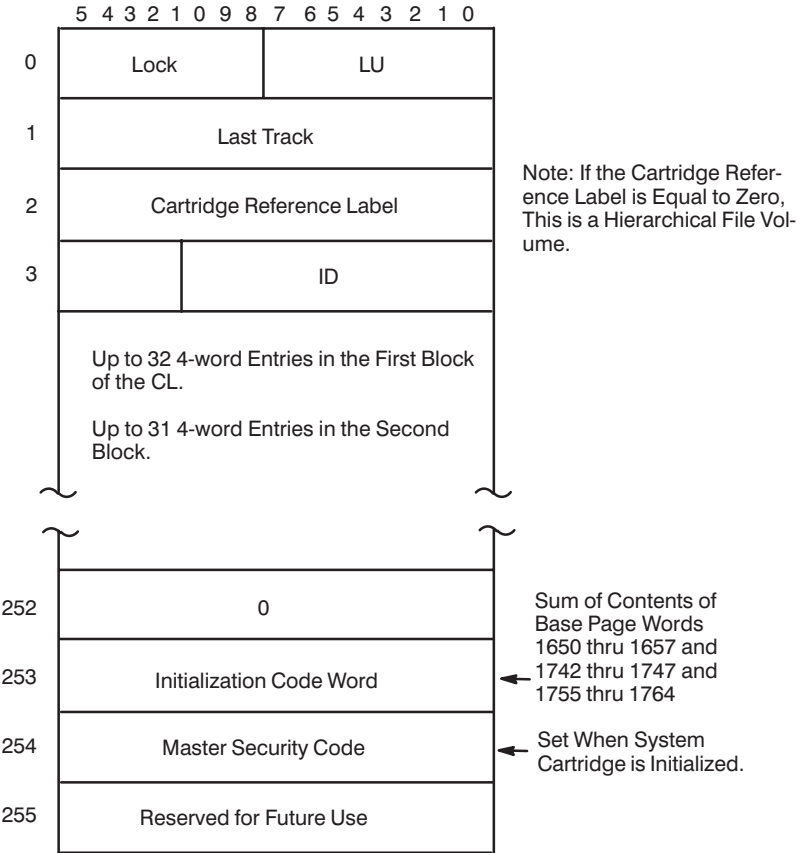
The standard values are shown within dark lines

Last FMGR error
Severity Code
Session Identifier
User's Capability Level

General Wait State Messages (State 3)

Message	Reason For Wait
LULK LU, LKPRG= progx	The listed program attempted to put a lock on logical unit lu. Program progx already has a lock on lu. The listed will be rescheduled when progx removes its lock.
RNxx, LKPRG= progx	The listed program attempted to set resource number xx. Program progx already has a lock on the resource number. The listed program will be rescheduled when progx removes the lock.
RESOURCE	The listed program attempted to allocate a resource number. The system has no more resource numbers available. The operating system will reschedule the listed program when a resource number is available.
CLASS #	The listed program requested a class number but the system has no more available. The operating system will reschedule the listed program when a class number becomes available.
CLxx	The listed program is waiting on completion of a class GET to class number xx.
prox	The listed program scheduled progx with wait. The listed program will be rescheduled when progx completes.
progx's QUEUE	The listed program scheduled progx on the queue with wait. Progx is not dormant so the listed program must wait. The listed program will be rescheduled after the scheduling of progx completes.
BL,EQT XX	Buffer limit exceeded on the controller in EQT entry xx.
EQLK xxx, LKPRG= PRGA	Program suspended for a locked EQT.
EQLK TABLE FULL	Program attempts to lock an EQT and the EQT table is full.

Cartridge Directory Format (Cartridge List)



Lock = 0 if not locked. Else is Keyword Table Offset of ID segment address of locking program.

Locked disks are available only to the locker.

ID identifies to whom the cartridge is mounted.

- ID = 0000 → Non-session
- ID = 7777 → System Cartridge
- 0 < ID < 7777 → Session Monitor Group or Private Cartridge

Note: Words 124, 125, 126 and 127 are unique only in the second block of the CL. The first block will hold 32 entries in words 0 through 127.

Error Codes

Account Error Codes	17-1
CI Error Messages	17-3
CLRQ Error Codes	17-3
Disk Allocation Error Codes	17-3
DS Error Codes	17-3
EXEC Call Error Codes	17-4
FMGR Error Codes	17-5
FMGR Unnumbered	17-8
FMP Error Messages	17-9
FORTRAN Runtime Error Codes	17-13
FTN7X Runtime Error Messages	17-13
FMP Errors (501–999)	17-17
I/O Errors (1000–1099)	17-17
DS Errors (1100–1199)	17-17
Other Runtime Errors	17-17
GASP Error Codes	17-18
Halt Errors	17-19
INDXR Error Messages	17-24
I/O Error Codes	17-25
Library Errors	17-27
Mathematical Subroutines	17-28
LOADR/MLLDR Error Codes	17-30
Logon Error Codes	17-33
LU Lock Error Codes	17-33
Outspool Error Messages	17-34
Parity Error Messages	17-34
Pascal Errors	17-35
READT/WRITT Error Codes	17-37
Reconfiguration Error Codes	17-39
Resource Number Errors	17-40
Schedule Call Error Codes	17-40
SMP Error Messages	17-41
System and Break-Mode Command Error Messages	17-42
Track (Disk Parity) Error Message	17-42
VMA/EMA Error Codes	17-43

Account Error Codes

ACCT-225	Session memory cannot be returned to system (reboot).
ACCT-223	Illegal shut down parameter.
ACCT-222	Illegal system LU.
ACCT-221	Not an active session.
ACCT-220	Corrupt station table spares.
ACCT-219	Not enough room in file for new table.
ACCT-218	Session not shut down.
ACCT-217	Illegal value for the number or depth of User-Definable Directory Search Paths (UDSPs). Values must be between 0 and 8; and if one is 0, both must be 0.
ACCT-216	Illegal response for primary program.
ACCT-215	List NAMR in transfer stack.
ACCT-213	Invalid memory request.
ACCT-212	Invalid number of SST spares.
ACCT-211	User or group ID not available.
ACCT-210	Conflict in SST definition.
ACCT-209	Invalid SST entry.
ACCT-208	Invalid disk limit.
ACCT-207	Invalid capability.
ACCT-206	Invalid file name.
ACCT-205	Invalid command.
ACCT-204	Invalid password.
ACCT-203	Invalid account name.
ACCT-202	Account with this name already exists.
ACCT-201	No free accounts.
ACCT-200	Account not found.
ACCT-099	An EXEC request made by D.RTR was aborted.
ACCT-046	Attempt to create extent 256. Make file size of main larger.
ACCT-041	No room in SST.
ACCT-040	LU not found in SST.
ACCT-039	Conflict in SST definition.
ACCT-035	Already 63 disks mounted to system.
ACCT-034	Disk already mounted.
ACCT-033	Not enough room on cartridge.

ACCT-032	Cartridge not found.
ACCT-030	Value too large for parameter.
ACCT-026	Queue full or max pending spools exceeded.
ACCT-025	No SPLCON room; the SPLCON is full.
ACCT-024	No more batch switches.
ACCT-023	No available pool files.
ACCT-022	No available pool LUs.
ACCT-021	Illegal destination LU.
ACCT-020	Illegal access LU.
ACCT-019	Illegal access on a system disk.
ACCT-018	Illegal LU; LU not assigned to system.
ACCT-017	Illegal read/write on type 0 file.
ACCT-016	Illegal type 0, or file blocksize=0.
ACCT-015	Illegal name.
ACCT-014	Directory full.
ACCT-013	Disk locked.
ACCT-012	EOF or SOF error.
ACCT-011	DCB not open.
ACCT-010	Not enough parameters.
ACCT-009	Attempt to use APOSN or force a type 0 file to type 1.
ACCT-008	File open or lock rejected.
ACCT-007	Illegal security code or illegal write on LU 2 or 3.
ACCT-006	File not found.
ACCT-005	Record length illegal.
ACCT-004	More than 32767 records in a type 2 file.
ACCT-003	Backspace illegal.
ACCT-002	Duplicate file name.
ACCT-001	Disk error.
ACCT 004	Illegal LU.
ACCT 012	LU not in session switch table.
ACCT 013	Transfer stack overflow.
ACCT 046	Insufficient capability.

CI Error Messages

CI error messages are self-explanatory. For further information, refer to the *RTE-6/VM CI User's Manual*, part number 92084-90036.

CLRQ Error Codes

CL01	Illegal class number or no class table in system.
CL02	Parameter or calling-sequence error.

Disk Allocation Error Codes

DR01	Not enough parameters were specified.
DR02	The number of tracks is zero, an illegal LU was specified, or number of tracks to release is zero or negative.
DR03	An attempt to release a track assigned to another program was made.

DS Error Codes

DS00	Local node is quiescent.
DS01	Communication line parity, protocol failure, 'STOP' received, cable disconnected, or other hardware error.
DS02	Communication line timeout error (DVA65 links only).
DS03	Illegal record size.
DS04	Illegal nodal address, node address not in nodal routine vector (NRV).
DS05	Request timeout.
DS06	Illegal request or monitor not active.
DS07	System table error.
DS08	Remote busy or resource unavailable.
DS09	Illegal or missing parameters.

EXEC Call Error Codes

DM Mapping error. An attempt was made to read/write outside of the mapped address space. The error message is:

```
DM VIOL = wwwwww (DMS violation register)
DM INST = xxxxxx (offending octal instruction code)
ABE pppppp qqqqqq r (A, B, and E-Registers)
XYO pppppp qqqqqq r (X, Y, and O-Registers)
MP progname 0
```

MP Memory protect error. The call is not an EXEC, \$LIBR, or \$LIBX call. The following message results:

```
MP INST = xxxxxx (offending octal instruction code)
ABE Pppppp qqqqqq r (A, B, and E-Registers)
XYO pppppp qqqqqq r (X, Y, and O-Registers)
MP progname O
```

The following errors have the same format as the DM or MP errors except that register contents are not reported.

- RE A reentrant subroutine attempted to call itself.
- RQ An illegal request code is specified in an EXEC call.
- TI A batch program exceeds the allowed time.

FMGR Error Codes

FMGR-768	Program aborted.
FMGR-105	D.RTR directory track buffer too small.
FMGR-104	Requested extent is missing.
FMGR-103	File directory is corrupt.
FMGR-102	Illegal D.RTR call sequence.
FMGR-101	Illegal parameter in D.RTR call.
FMGR-099	Directory manager EXEC request was aborted.
FMGR-052	Spool shut down. Spool file setup failed.
FMGR-049	Copy verify failed.
FMGR-048	Spool not initialized or SMP cannot be scheduled.
FMGR-047	No session LU available for spool file.
FMGR-046	Greater than 255 extents.
FMGR-041	No room in SST.
FMGR-040	LU not found in SST.
FMGR-039	Spool LU not mapped to the spool driver.
FMGR-038	Illegal scratch file number.
FMGR-037	Attempt to purge an active type 6 file.
FMGR-036	Lock error on device.
FMGR-035	Already 63 disks mounted to system.
FMGR-034	Disk already mounted.
FMGR-033	Not enough room on cartridge.
FMGR-032	Cartridge not found.
FMGR-030	Value too large for parameter.
FMGR-026	Queue full or max pending spools exceeded.
FMGR-025	No SPLCON room.
FMGR-024	No more batch switches
FMGR-023	No available spool files.
FMGR-022	No available spool LUs.
FMGR-021	Illegal destination LU.
FMGR-020	Illegal access LU.
FMGR-019	Illegal access on a system disk.
FMGR-018	Illegal LU.
FMGR-017	Illegal read/write on type 0 file.
FMGR-016	Illegal type 0 or size=0.
FMGR-015	Illegal name.

FMGR-014	Directory full.
FMGR-013	Disk locked.
FMGR-012	EOF or SOF error.
FMGR-011	DCB not open.
FMGR-010	Not enough parameters.
FMGR-009	Attempt to use APOSN or force to 1 a type 0 file.
FMGR-008	File open or lock rejected.
FMGR-007	Illegal security code or illegal write on LU 2 or 3.
FMGR-006	File not found.
FMGR-005	Record length illegal.
FMGR-004	Record size of type 2 file is 0 or undefined.
FMGR-003	Backspace illegal.
FMGR-002	Duplicate file name.
FMGR-001	Disk error, the disk is down.
FMGR 000	Break, informative message only; no error has occurred.
FMGR 001	Disk error; LU reported, disk associated with the LU is down.
FMGR 002	Initialize LU 2!
FMGR 003	Initialize LU 3!
FMGR 004	Illegal response to FMGR 002 or FMGR 003.
FMGR 005	Required track not available; relative TAT position reported.
FMGR 006	FMGR suspended.
FMGR 007	Checksum error.
FMGR 008	D.RTR not loaded.
FMGR 009	ID segment not found.
FMGR 010	Input error.
FMGR 011	Do "OF,XXXXX,8" on named programs.
FMGR 012	Duplicate disk label or LU.
FMGR 013	TR stack overflow.
FMGR 014	Required ID segment not found.
FMGR 015	LS track report.
FMGR 016	Insufficient system tracks for RP.
FMGR 017	ID segment not set up by RP.
FMGR 018	Program not dormant.
FMGR 019	File not set up by SP on current system.
FMGR 020	Illegal type 0 file.

FMGR 021	Illegal disk specified.
FMGR 022	Copy terminated.
FMGR 023	Duplicate program name.
FMGR 038	Attempt to purge active file.
FMGR 041	Program cannot be a segment.
FMGR 042	LU cannot be switched.
FMGR 043	LU not found in SST.
FMGR 044	No messages waiting.
FMGR 045	Session command only.
FMGR 046	Insufficient capability.
FMGR 047	Spool set up failed.
FMGR 048	Global set out of range.
FMGR 049	Cannot run RP'ed program or partition too small.
FMGR 050	Not enough parameters.
FMGR 051	Illegal master security code
FMGR 052	Illegal LU.
FMGR 053	Illegal label or ilabel.
FMGR 054	Disk not mounted.
FMGR 055	Missing parameter.
FMGR 056	Bad parameter.
FMGR 057	Bad track not in file area.
FMGR 058	LG area empty.
FMGR 059	Reported track unavailable.
FMGR 060	Do you really want to purge this disk?
FMGR 061	Do a "DC" and a "MC" on this CR.
FMGR 062	More than 63 disks.
FMGR 063	Exceeding session disk limit.
FMGR 064	No disk available from disk pool.
FMGR 065	Conflict in SST definition.
FMGR 066	No room in SST.
FMGR 067	Program not found.
FMGR 068	LU not in variable part of SST.
FMGR 069	Job LOGON failed.
FMGR 070	Sectors/track value too large.
FMGR 071	Do "EX,SP" to save or "EX,RP" to release private cartridges.
FMGR 072	LU not interactive.

FMGR 073	Account not found.
FMGR 074	JO command expected.
FMGR 075	Cannot restore type 6 PGM file (user protected).
FMGR 076	Cannot restore type 6 PGM file (group protected).
FMGR 077	Cannot restore type 6 PGM file (insufficient capability).
FMGR 078	Cannot restore type 6 program file (internal error).
FMGR 079	Warning; records truncated to 128 words.
FMGR 080	Cannot find named EMA.

FMGR Unnumbered

Error Message	Meaning
ABEND OPERATOR	The job has been aborted by operator request, or has been aborted because of spool I/O error.
JOB xxxx ABORTED	Error encountered during job execution.
ABEND EOJ IN ssssss	An :EO or :JO command was encountered, but in a different level from the original :JO command. For example, control has transferred from PROG1 to PROG2. PROG2 contains :EO or :JO command. ssssss is the file name or LU number where :EO or :JO occurred.
ABEND JOB LIMIT	The job time limit (set via the :JO command) has been exceeded.
ABEND RUN LIMIT	The run time limit (set via the :TL command) has been exceeded.
FMGR WAITING ON LU xx	LU xx is down or locked to another program.

FMP Error Messages

Error	Error Message
-000	(no error)
-001	DISK ERROR
-002	FILE ALREADY EXISTS
-003	BACKSPACE ILLEGAL
-004	RECORD SIZE ILLEGAL
-005	BAD RECORD LENGTH
-006	NO SUCH FILE
-007	INCORRECT SECURITY CODE
-008	FILE IS ALREADY OPEN
-009	MUST NOT BE A DEVICE
-010	NOT ENOUGH PARAMETERS
-011	DCB IS NOT OPEN
-012	ILLEGAL FILE POSITION
-013	DISK IS LOCKED
-014	DIRECTORY IS FULL
-015	ILLEGAL NAME
-016	SIZE=0 OR TYPE 0 FILE ACCESS
-017	DEVICE I/O FAILED
-018	ILLEGAL LU.
-019	ILLEGAL LU 2 OR 3 ACCESS
-020	ILLEGAL ACCESS LU
-021	ILLEGAL DESTINATION LU
-022	NO AVAILABLE SPOOL LU
-023	NO AVAILABLE SPOOL FILES
-024	NO MORE BATCH SWITCHES
-025	NO SPLCON ROOM
-026	QUEUE FULL OR TOO MANY PENDING SPOOLS
-030	VALUE TOO LARGE FOR PARAMETER
-032	NO SUCH CARTRIDGE
-033	RAN OUT OF DISK SPACE
-034	DISK IS ALREADY MOUNTED
-035	ALREADY 63 DISKS MOUNTED TO SYSTEM
-036	LOCK ERROR ON DEVICE

-037	PROGRAM IS ACTIVE
-038	ILLEGAL SCRATCH FILE NUMBER
-039	SPOOL LU NOT MAPPED TO SPOOL DRIVER.
-040	LU NOT FOUND IN SST
-041	NO ROOM IN SST
-046	MORE THAN 255 EXTENTS
-047	NO SESSION LU AVAILABLE FOR SPOOL FILE
-048	SPOOL NOT INITIALIZED
-049	COPY VERIFY FAILED
-050	NO FILES SELECTED
-051	DIRECTORY IS EMPTY
-052	SPOOL SHUT DOWN. SPOOL FILE SETUP FAILED
-053	PROGRAM ASSIGNED TO BAD PARTITION
-054	PARTITION TOO SMALL FOR PROGRAM
-055	NO ROOM IN SHAREABLE EMA TABLE
-056	SHEMA ASSIGNED TO NON-EXISTENT PARTITION
-057	PARTITION TOO SMALL FOR SHAREABLE EMA
-058	PROGRAM OR DATA ASSIGNED TO SHEMA PARTITION
-059	63 PROGRAMS USING SHAREABLE EMA AREA
-099	D.RTR EXEC REQUEST ABORTED!
-101	ILLEGAL PARAMETER IN D.RTR CALL
-102	D.RTR NOT AVAILABLE
-103	DIRECTORY IS CORRUPT
-104	MISSING EXTENT
-105	D.RTR NOT SIZED UP
-108	ILLEGAL NUMBER OF SECTORS/TRACK
-200	NO WORKING DIRECTORY
-201	DIRECTORY NOT EMPTY
-202	DID NOT ASK TO READ
-203	DID NOT ASK TO WRITE
-204	FILE READ PROTECTED
-205	FILE WRITE PROTECTED
-206	DIRECTORY READ PROTECTED

-207	DIRECTORY WRITE PROTECTED
-208	DUPLICATE DIRECTORY NAME
-209	NO SUCH DIRECTORY
-210	UNPURGE FAILED
-211	DIRECTORIES NOT ON THE SAME LU
-212	CANNOT CHANGE THAT PROPERTY
-213	TOO MANY OPEN FILES
-214	DISK IS NOT MOUNTED
-215	TOO MANY DIRECTORIES
-216	YOU DO NOT OWN
-217	BAD DIRECTORY BLOCK
-218	MUST SPECIFY AN LU
-219	NO REMOTE ACCESS
-220	DSRTR NOT AVAILABLE
-221	FILES ARE OPEN ON LU
-222	LU HAS OLD DIRECTORY
-223	ILLEGAL DCB BUFFER SIZE
-224	NO FREE ID SEGMENTS
-225	PROGRAM IS BUSY
-226	PROGRAM ABORTED
-227	PROGRAM DOESN'T FIT IN PARTITION
-228	NO SAM TO PASS STRING
-229	ACTIVE WORKING DIRECTORY OR A UDSP POINTS TO THIS DIRECTORY
-230	ILLEGAL USE OF DIRECTORY
-231	STRING IS TOO LONG
-232	UNKNOWN FOR FMGR FILE
-233	NO SUCH USER
-234	SIZE MISMATCH ON COPY
-235	BREAK FLAG DETECTED
-236	YOU ARE NOT A SUPERUSER
-237	MUST NOT BE REMOTE
-238	ILLEGAL PROGRAM FILE
-239	PROGRAM NAME EXISTS
-242	DISK I/O FAILED
-243	PARAMETER ERROR
-244	MAPPING ERROR
-246	SYSTEM COMMON CHANGED

-247	UDSP NOT DEFINED
-248	INVALID DIRECTORY ADDRESS FOUND
-250	D.ERR NOT AVAILABLE
-252	DISK LU IS DOWN
-253	DISK LU IS LOCKED
-254	NO SUCH GROUP
-256	NO SUCH SESSION
-257	NO SUCH PROGRAM
-270	UPDATE TIME ALREADY CURRENT
-300	ILLEGAL REMOTE ACCESS
-301	TOO MANY REMOTE CONNECTIONS
-302	NO SUCH NODE
-303	TOO MANY SESSIONS
-304	NO SUCH ACCOUNT
-305	INCORRECT PASSWORD
-306	CAN'T ACCESS ACCOUNT
-308	CONNECTION BROKEN
-310	DS IS NOT INITIALIZED
-311	DS LINK IS NOT CONNECTED
-312	REMOTE SYSTEM DOESN'T RESPOND
-313	NO TRFAS AT REMOTE SYSTEM
-315	DS ERROR DSXX(X), NODE YY
-401	MESSAGE NUMBER NOT FOUND
-402	SPECIFIED BUFFER OVERFLOW

FORTRAN Runtime Error Codes

During execution of an object program, error messages may be printed on the output unit by the input/output system supplied for FORTRAN programs. The error message is printed in the form:

```
program name *RUNTIME ERROR* nnnn @ xxxxx
```

where:

program name is the name of the user program.

nnnn is the error code.

xxxxx is the approximate logical address of the statement that caused the error.

If the IOSTAT=ios and ERR=label are present, the input/output error code is stored in ios and control transfers to label, where a user routine can decode and handle the error, if desired.

FTN7X Runtime Error Messages

IOSTAT

Error Condition Meaning

-1	An EOF was read on a sequential file, or the end of an internal file was reached on a READ or WRITE.
450	Invalid FORTRAN unit number (less than zero) or system unit number (greater than 255).
451	Unrecognized STATUS value; legal values in an OPEN statement are OLD, NEW, SCRATCH and UNKNOWN.
452	No file name (FILE=) given; file names are required when STATUS is OLD or NEW.
453	File name (FILE=) supplied; no file names are allowed when STATUS is SCRATCH.
454	Unrecognized ACCESS value; legal values are SEQUENTIAL and DIRECT.
455	Unrecognized FORM value; legal values are FORMATTED and UNFORMATTED.
456	The value for MAXREC, RECL and/or BUFSIZE is negative or zero. These parameters must have positive values. If all values seem correct, and maxrec is used, ask the System Manager to verify that the run-time library is not an older version that predates the current compiler.

- 457 Unrecognized BLANK value; legal values are NULL and ZERO.
- 458 The maximum number of scratch files, 99, were in use, so this OPEN of another scratch file could not be done.
- 459 This file has already been opened and connected to a different unit; a file can be connected to only one unit at a time.
- 460 The OPEN specified direct access, but the file to be opened was sequential access (not type 1 or 2).
- 461 The OPEN specified sequential access, but the file to be opened was direct access (type 1 or 2).
- 462 The file was not found, and STATUS was OLD. Same as error number 506.
- 463 Unrecognized STATUS value; legal values in a CLOSE statement are KEEP DELETE.
- 464 An ENDFILE was attempted on a direct-access file; such files do not have EOFs as such, and ENDFILE is illegal.
- 465 Invalid file name given in FILE=. Same as error number 515.
- 466 All connections specified in the \$FILES are in use; no more OPENS can be done until a CLOSE is done.
- 467 All disk file connections specified in the \$FILES are in use; no more disk file OPENS can be done until a disk file CLOSE is done.
- 468 The record length given in RECL does not match the actual record length of the file.
- 469 The file position given in FFLOC is odd; only even byte positions are allowed, because the file system cannot position a file to an odd byte position.
- 470 Unrecognized USE value; legal values are EXCLUSIVE, NONEXCLUSIVE and UPDATE.
- 471 The system unit number used is not accessible from this program; that is, it is not in the session SST.
- 472 Could get neither read nor write permission for this file.
- 473 (not currently used)
- 474 A record number (REC=) was supplied in the READ or WRITE, but the unit was not connected to a direct access file.

- 475 A RECL value must/must not be supplied.
RECL must be used if, and only if, ACCESS is
DIRECT.
- 476 (not currently used)
- 477 A NODE value was supplied (other than -1),
but the \$FILES directive did not contain the DS
keyword.
- 478 An OPEN of a unit/file that was already
connected tried to change attributes other than
BLANK.
- 479 An OPEN was tried with \$FILES 0,0; or the
library routines to support OPEN were not
loaded correctly.
- 480 A CLOSE was tried with \$FILES 0,0; or the
library routines to support CLOSE were not
loaded correctly.
- 481 An INQUIRE was tried with \$FILES 0,0; or the
library routines to support INQUIRE were not
loaded correctly.
- 482 The library routines to support BACKSPACE,
ENDFILE and REWIND were not loaded
correctly.
- 483 This INQUIRE statement tried to inquire about
a disk file, but the \$FILES directive did not
specify any disk connections. At least one disk
connection must be specified, even if OPEN, or
a disk file will not be done.
- 484 OPEN statement tried to open a disk file, but
the \$FILES directive did not specify any disk
connections.
- 485 This OPEN statement specifies a direct access
file or uses RECL, but the \$FILES directive did
not specify any disk connections.
- 486 Attempt to use DNODE, which is illegal in
FORTRAN 77 programs. Use an OPEN
statement to connect to a remote unit.
- 487 ZBUF, ZLEN or secondary/tertiary address
supplied in this READ or WRITE statement,
but unit is connected to a disk file.
- 488 REC supplied in this READ or WRITE
statement is negative.
- 489 (not currently used)
- 490 (not currently used)
- 491 This FORMAT has an invalid field width (w),
number of digits (d), minimum number of digits
(m), or size of exponent output(e).

- 492 This FORMAT does not begin with a left parenthesis, or has too many levels of parentheses.
- 493 Unrecognized format character, or use of a negative value in a format (except scale), or no conversions given in the format, but I/O list was not empty.
- 494 Illegal character in a numeric input field.
- 495 Numeric input field has an ill-formed number or logical value, or an octal number was too large.
- 496 Discrepancy in record size and/or I/O list length and/or internal buffer size. Could be due to:
- a. Input record (or amount that fits in internal buffer) was not large enough to satisfy an unformatted READ list. If unknown-length records are being read with an unformatted READ, the error can be trapped with IOSTAT and the actual record length recovered using ITLOG.
 - b. Output record, as specified in unformatted WRITE list or a FORMAT, was too large to fit in the internal buffer. See library routine LGBUF.
 - c. Output record, as specified in unformatted WRITE list or a FORMAT statement, was larger than the record size for the direct-access file to which this unit is connected.
- 497 A format specifier is illegal for the type of the list item that matches it.

FMP Errors (501 – 999)

Runtime errors in the range of 501 to 999 are FMP errors, except for 600 and 601.

- 600 Left and right side of character assignment overlap.
- 601 Substring out of bounds.

Otherwise, see FMP Error Codes.

Mapping:

Runtime error is 500 – FMP error code. To determine FMP error, subtract runtime error from 500. For example, runtime error 725 is FMP error –225.

I/O Errors (1000 – 1099)

See I/O Error Codes.

Mapping:

Runtime error is 1000 + xx, where the I/O error is IOxx. For example, runtime error 1005 represents an IO05 error.

DS Errors (1100 – 1199)

See DS Error Codes.

Mapping:

Runtime error is 1100 + xx, where the DS error is DS xx. For example, runtime error 1109 represents a DS09 error.

Other Runtime Errors

- XX99 Partially malformed error messages.
- 1199 All IO and DS errors where the form is not IOXX or DSXX and which are not FMP errors are mapped to 1999.

GASP Error Codes

GASP-48	Spooling not initialized or SMP cannot be scheduled.
GASP-33	Not enough room on disk cartridge.
GASP-32	Disk cartridge not found.
GASP-14	Directory full.
GASP-13	Disk locked.
GASP-12	EOF or SOF error.
GASP-8	File open or lock rejected.
GASP-7	Illegal security code or illegal write on LU 2 or 3.
GASP-6	File not found or no room to create spool files.
GASP-4	More than 32767 records in a type 2 file.
GASP-2	Duplicate file name.
GASP-1	Disk error, disk is down.
GASP 1	Disk associated with LU NN is down.
GASP 2	Number out of range.
GASP 3	Bad job number!
GASP 4	Illegal status.
GASP 5	Illegal command.
GASP 6	Not found. Specified job or spool not currently assigned.
GASP 7	GASP segment not found.
GASP 43	LU not found in SST.
GASP 46	Insufficient capability.
GASP 54	Spool cartridge not mounted.
GASP 55	Missing parameter.
GASP 56	Bad parameter.

Halt Errors

Octal Halt Code	Meaning
0	Table area I wrap-around halt. This halt is located in the module \$TB1. Wrap-around halts are meant to stop the processor if it is executing data.
2	Memory wrap-around halt, in location 2 of the system map, placed there by RTIOQ at system bootup. May indicate possible hardware problem.
3	Memory wrap-around halt, in location 3 of the system map, placed there by RTIOQ at system bootup. May indicate possible hardware problem.
4	Powerfail occurred and one of the following is true: <ul style="list-style-type: none">– The powerfail/autorestart subsystem is not installed.– Powerfail code did not execute completely.– Powerfail occurred during system bootup.
5	<p>If the halt code is a 102005B, a parity error has occurred in the system map or the user area containing system tables. The A-Register contains the physical page number and the B-Register contains the logical parity error address. Issued by the parity error handler PERR6.</p> <p>If the halt code is a 103005B, a parity error has occurred during a DCPC transfer while the system map was enabled. Issued by PERR6.</p>
6	A memory partition was not properly linked into an operating system partition list. Issued by the dispatcher in OS4DP. May indicate a corrupt operating system.
10	FMGR or D.RTR cannot be scheduled at system bootup because there is no partition large enough. Issued by the scheduler in OS3SC. Perform a reconfiguration boot and redefine memory to include large enough partitions. As of Revision 5000, D.RTR requires 36 pages.

- 11 A disk error occurred while the disk loader ROM was trying to read in the boot extension. Reboot the system to restart the read. If it still fails, one of the following may be true:
- An incorrect loader ROM was specified in the S-register.
 - The boot extension does not reside on the disk subchannel specified in the S-register. Verify that the correct disk unit and head number were stored in the S-register.
 - A hardware problem exists with the system disk. Run the diagnostics to isolate the problem.
- 20 Uninstalled memory halt. At bootup, the system verifies that all the memory declared is installed. This halt is issued by the scheduler in the OS3SC module. The uninstalled page number will be displayed in the A- and B-Registers. Reboot and reconfigure memory.
- 21 Bad or missing RTE-6/VM firmware. The firmware did not pass self-test issued at system bootup in SCHED6 at \$STRT. The self-test for the OS microcode is 105355B and the VMA self-test instruction is 105242B. The operating system detects self-test failures based on where the microcode returns (P+1 = failure).
- 22 Issued by the configurator module \$CNF3 at system bootup if one of the following conditions is true:
- The configurator extension (\$CNFX) is not a type 3 program.
 - No ID segment can be found for \$CNFX.
 - There is no partition with 3 contiguous pages of good memory.
- 30 Error was encountered in the disk I/O process by the disk loader ROM. If the disk is a MAC disk, then the B-Register contains status word 1 and the A-Register contains status word 2. The two status words for MAC drives are defined in the 13037 disk controller manual, part number 13037-90006. A common cause is that the disk has not yet attained the correct speed. Halt 30 is not a valid halt code for the CS/80 disk loader ROM.

- 31 This halt is issued by the boot extension code and is similar to a halt 30 in that the cause is a disk I/O problem. For MAC and ICD disks, the B-Register contains status word 1 and the A-Register contains status word 2.
For CS/80 disks the A-Register contains either 1, 2, or 3 (1 = reject error, 2 = fault error, and 3 = access error). The B-Register contains the error status word as described in the DVM33 reference manual, part number 92084-90025.
For both halt 30 and 31 the recommended action is to retry the bootup procedure. If the problem persists, run the disk diagnostics.
- 33 One of the following conditions was detected by the Multi-level Segmentation (MLS) load on call routine \$LOC:
- The program that called \$LOC does not have MLS flag set in ID segment word 33.
 - The program has no memory resident nodes.
 - While setting up the map registers for a memory resident node, the \$LOC routine detected that the node location in memory is invalid.
- 41 Negative count for TDB move or TDB address is greater than 32K. This halt was removed from the current version. (Revision 5000 or greater) of OS2EX in favor of an MP error.
- 47 An attempt was made to enter one of the operating system code partitions before it was mapped into physical memory. The configurator module (\$CNF1) loads the six operating system code partitions into memory at system bootup. The most probable cause of this halt is that the system disk driver encountered an error and tried to call an error handling routine located in one of the code partitions. The MAPOS module detects that the code is not yet loaded into memory and issues this halt in the OSMAP routine. The recommended action is to examine your answer file to ensure that the system disk is correctly defined.
- 50 The fourth word of the shareable EMA table entry for a SHEMA partition indicates the number of active users is less than zero. The dispatcher checks for this condition in the ABORT routine of DISP6.

- 51 A program is assigned to a shareable EMA partition or to a partition that is a subpartition of a shareable EMA partition. This is detected in the \$FPTN routine which is called by the dispatcher to find a partition for a disk resident program to run in. This halt occurs if bit 13 of the fifth word of the MAT entry is set.
- 52 Similar to a halt 51 except that it occurs if bit 12 of the fifth word of MAT entry is set. Indicates that the partition is currently in use as a SHEMA partition.
- 53 This halt is issued by the \$FPTN of OS4DP if a mother partition is found having the chain bit set and no program is resident.
- 54 A partition that should have been empty had a program resident. This condition is detected in OS4DP.
- 55 Halt 55 can result from any of the following conditions:
- The loader ROM code for the MAC, ICD, and CS/80 disks contains a 102055B (halt 55). This is used as the starting address for the boot extension code. When this halt occurs in the loader ROM, the boot-up procedure should be tried again, and if it persists, the firmware diagnostics should be run. This halt is issued by the loader ROM if a slow boot is performed and the halt 55 occurs before the halt 77.
 - The configurator module \$CNF1 could not find an EQT with the equipment type code of the console. In this case, the halt 55 occurs after the halt 77 when a slow boot is performed.
 - The dispatcher module OS4DP found a partition defined in the shareable EMA table (\$EMTB) but word 4 of the memory allocation table does not have the shareable EMA bit set for this partition.
- 56 The dispatcher module OS4DP has determined that a program being dispatched needs a subpartition to run and that subpartition is currently in chained mode (the mother partition is in use). When the dispatcher attempts to swap out the resident of the mother partition, the memory allocation table is found to be corrupt.

- 57 A mother partition is occupied, but the subpartitions were not in chained mode. This occurs in OS4DP.
- 60 A program was found to reside in a shareable EMA partition that should have been empty. This halt is not active as of Revision 2340 in OS4DP.
- 61 The last shareable EMA program using a partition has terminated (the active count = 0) and the SHEMA partition is not locked. When the status field in the Memory Allocation Table was examined, a status of 1 was found, indicating a program is resident. This condition is detected in OS4DP.
- 63 The dispatcher swap out routine was called and the partition was found to be in use as a SHEMA partition. This routine is located in OS4DP.
- 75 System debug halt. Setting the S-Register to 70000B after a slow boot results in a halt 75 which enables you to pass the terminal select code to the system debugger.
- 76 Microcode troubleshooting halt. Setting the S-Register to 60000B after a slow boot results in halt 76 which allows the modification of the microcode self-test.
- 77 Halt 77 is typically used to indicate the successful completion of an operation. For example, a halt 77 during slow boot indicates that part one of the operating system was successfully loaded into memory.
- 105242, 105355 These codes appear in the T-Register on bootup. They indicate self-test errors. See HLT 21 above.

INDXR Error Messages

Error Message

??

*** FMP -002 FILE DESCRIPTOR,
OVERLAY (Y OR N)? ***

*** INPUT ERROR,
INDXR ABORTED ***

*** ONLY ONE LEVEL OF
TRANSFER ALLOWED ***

*** INDEXED FILE ALREADY EXISTS,
REQUEST IGNORED ***

*** FILE MUST BE TYPE 5,
REQUEST IGNORED ***

*** CHECKSUM ERROR,
INDXR ABORTED ***

*** DIRECTORY TOO LARGE,
INDXR ABORTED ***

*** LIST FILE MUST BE TYPE 3
OR 4 ***

Meaning

An invalid command was entered, try again.

INDXR has found in response to a CR or LI command that the file specified already exists.

An invalid transfer file was specified in the runstring.

Only one transfer file may be opened for command input to this utility.

An attempt was made to use the CReate command more than once.

The file specified in the INdEx command was not a type 5 (relocatable) file.

While reading records from the file specified by the last INdEx command, a checksum error was detected in one of the records.

The INDXR creates a scratch file in which to build the directory index. This file is created on the first cartridge in the cartridge list. If this file creates more than 255 extents or fills up the remainder of the cartridge, then this error is returned.

An already existing FMP file used as a list file must be a type 3 or 4.

*** LIST FILE/LU ALREADY OPEN,
REQUEST IGNORED ***

This list file may only be opened/created once. If the LI[ST] command is invoked more than once then this error is issued.

*** FROM/TO FILES MUST BE
DIFFERENT ***

The same file was specified in the CR and IN commands. The IN command will be ignored.

*** INDEXED FILE NOT OPEN,
REQUEST IGNORED ***

An IN command was entered before a CR command.

I/O Error Codes

IO00	An illegal class number was specified. Outside table, not allocated, or bad security code.
IO01	Not enough parameters were specified.
IO02	An illegal LU number was specified.
IO03	Illegal EQT referenced by LU in I/O call (select code=0).
IO04	An illegal user buffer was specified. Extends beyond RT/BG area or not enough System Available Memory (SAM) to buffer the request.
IO05	An illegal disk track or sector was specified.
IO06	A reference was made to a protected track or to unassigned LG tracks.
IO07	The driver has rejected the call.
IO09	The LG tracks overflowed.
IO10	Class GET call issued while one call already outstanding.
IO11	A type 4 program made an unbuffered I/O request to a driver that did not do its own mapping.
IO12	An I/O request specified an LU not defined for use by this session. The SL command reports all LUs available to your session, or insufficient capability to directly access an LU.
IO13	An I/O request specified an LU that was either locked to another program, or pointed to an EQT that was locked to another program.

IO14	An I/O request was issued with the no-suspend option.
IO15	Buffer size of a type 6 program is greater than what will fit in the user map.
IO16	CPU backplane failure or I/O extender timing failure.
IO20	Read attempted on write only spool file.
IO21	Read attempted past end-of-file.
IO22	Second attempt to read JCL card from batch input file by other than FMGR. Revise program and rerun.
IO23	Write attempted on read only spool file.
IO24	Write attempted beyond end-of-file; usually, spool file overflow.
IO25	Attempt to access spool LU that is not currently set up.
IO26	I/O request made to a spool that has been terminated by the GASP KS command.
ILL INT	An illegal interrupt occurred on the specified channel.

The following error message format is used to report I/O errors:

```

NR
IO ET L xxx E yyy S zz qqg
TO
PE

```

where:

```

xxx   Device's LU.
yyy   Device's EQT.
zz    Device's subchannel.
qqg   Device status returned by driver (if the
       driver is down at I/O request the
       status=***).

```

IOET	An end-of-tape condition occurred on the specified LU.
IONR	The specified LU is not ready. Make the device ready and set the EQT up.
IOTO	The specified LU has timed out.
IOPE	A parity error occurred in the data transmission from the specified LU.

Library Errors

Error messages have the following format:

```
/pname *RUNTIME ERROR* xxxx @ yyyyy
```

where:

pname	is the name of the program where the error was encountered.
xxxx	is the error code; it can take one of three forms:
nnnn	4-digit numeric error code
EOF	end-of-file error
nnaa	intrinsic routine error
nn	= routine number
aa	= OF integer or floating point overflow UN operation is undefined for this argument, for example, $\log(-1)$ OR operation is defined but is computationally unfeasible for this argument, for example, $\sin(1e22)$
yyyyy	is the address within the program at which the error occurred. The module and line number can be determined using a load map and a compilation listing with the Q option.

Mathematical Subroutines

Error Message	*Expression Used in Program	Error Condition
02-UN	LOG(r)	$r \leq 0$
	ALOG(r)	$r \leq 0$
	LOG10(r)	$r \leq 0$
	ALOG10(r)	$r \leq 0$
	LOG(c)	$c = (0,0)$
	CLOG(c)	$c = (0,0)$
	LOG(d)	$d \leq 0$
	DLOG(d)	$d \leq 0$
	LOG10(d)	$d \leq 0$
	DLOG10(d)	$d \leq 0$
03-UN	SQRT(r)	$r < 0$
	DSQRT(x)	$x < 0$
	DSQRT(d)	$d < 0$
04-UN	r**r	base=0, exponent ≤ 0 or base<0,exponent #0
05-OR	SIN(r)	r or real(c)
	COS(r)	outside
	CSIN(c)	$[-8192*\pi, 8191.75*\pi]$
	CCOS(c)	
	CEXP(c)	
	DSIN(d) DCOS(d)	d outside $[-2**23, +2**23]$
06-UN	r**i	base=0,exponent ≤ 0
06-OR	r**j	exponent outside $[-32768, +32767]$
07-OF	EXP(r)	r,d or real(c)
	DEXP(d)	> 88.03
	EXP(c)	
	r**r	overflow
	r**d	
	d**r	
d**d		
08-UN	i**i	base=0, exponent ≤ 0
	i**j	
	j**i	
	j**j	

Parameter types are as follows:

- r = REAL*4
- x = EXTENDED PRECISION (REAL*6)
- d = DOUBLE PRECISION (REAL*8)
- i = INTEGER*2
- j = DOUBLE INTEGER (INTEGER*4)
- c = COMPLEX*8 or COMPLEX*16, (real(c), imag(c))

Error Message	*Expression Used in Program	Error Condition
08-OF	$i^{**}i, i^{**}j$ $j^{**}i, j^{**}j$	overflow
09-OR	TAN(r) DTAN(x) DTAN(d)	r or x outside [-8192*PI, +8191.75*PI] d outside [-2**23,+2**23]
10-OF	DEXP(x) $x^{**}x$ $x^{**}r$ $r^{**}x$	$x > 88.03$ overflow
11-UN	DLOG(x) DLOG10(x)	$x \leq 0$
12-UN	$x^{**}i$ $d^{**}i$	base=0, exponent ≤ 0
13-UN	$x^{**}x$ $x^{**}r$ $r^{**}x$ $r^{**}d$ $d^{**}r$ $d^{**}d$	base<0 or base=0, exponent ≤ 0
14-UN	$c^{**}i$	base=(0,0),exponent ≤ 0
15-UN	DATAN2 (d1,d2)	d1=d2=0
21-UN	ASIN(r)	$r > 1$
22-UN	ACOS(r)	$r > 1$
23-OR	SINH(r) CSIN(c) CCOS(c)	$r > 88.722839$ imag(c) > 88.722839
24-OR	COSH(r)	$r > 88.722839$
26-UN	ACOSH(r)	$r < 1$
27-UN	ATANH(r)	$r \geq 1$
31-UN	DASIN(d)	$d > 1$
32-UN	DACOS(d)	$d > 1$
33-OR	DSINH(d)	$d > 88.722839$
34-OR	DCOSH(d)	$d > 88.722839$
36-UN	DACSH(d)	$d < 1$

Parameter types are as follows:

- r = REAL*4
- x = EXTENDED PRECISION (REAL*6)
- d = DOUBLE PRECISION (REAL*8)
- i = INTEGER*2
- j = DOUBLE INTEGER (INTEGER*4)
- c = COMPLEX*8 or COMPLEX*16, (real(c), imag(c))

Error Message	*Expression Used in Program	Error Condition
37-UN	DATNH(d)	d > 1
41-OR	CTAN(c)	real(c) outside [-4096*pi, +4095.875*pi]

Utility Subroutines:

Subroutine	Error
MAGTP	Returns on an illegal call.
.SWCH	Returns if element is out of range.

LOADR/MLLDR Error Codes

L-CK SUM	This is a checksum error. A file specified to the loader that did not contain relocatable format code.
L-CM BLK	This is a common block error.
L-CO RES	Attempt to replace or purge a memory-resident program. This is illegal.
L-DU ENT	Duplicate entry point.
L-DU PGM	Attempt to load the same program several times without getting rid of the earlier loads.
L-EX CPY	Attempt to replace or purge a program where copies of that program existed.
L-ID EXT	No ID extensions available for the EMA program.
L-IL ALC	External references to named COMMON that appear in an allocate record are not allowed before the allocate record occurs.
L-IL CMD	Attempt to purge a program under batch or attempt to use the PU command within a loader command file.
L-IL DRN	Illegal disk-resident node specification.

Parameter types are as follows:

- r = REAL*4
- x = EXTENDED PRECISION (REAL*6)
- d = DOUBLE PRECISION (REAL*8)
- i = INTEGER*2
- j = DOUBLE INTEGER (INTEGER*4)
- c = COMPLEX*8 or COMPLEX*16, (real(c), imag(c))

L-IL EMA	Tried to use shareable EMA with the old EMA relocatable format.
L-IL MLS	This error occurs during preliminary checking of the command file, or when an instruction other than a JSB is used to access a symbol in a son node during load.
L-IL PRM	The runstring or a command in a command file contained an error.
L-IL PTN	A partition specified in the load of the program does not exist or has been downed due to a parity error.
L-IL REC	The loader found a record that was not a NAM, ENT, EXT, DBL, EMA, GEN, LOD, END record, or extended record. The checksum was OK but the record was not identified.
L-IL REL	The compiler produced an illegal record.
L-IL RPL	Tried to do a JSB to a user-specified RPL in a son node.
L-IL SCB	Illegal session control block value (negative capability level).
L-IL SEG	Illegal segment specification.
L-IN CAP	Attempt to load, purge, or replace a permanently loaded program without having a session capability level high enough to perform this function.
L-LM LIB	The limit on the number of libraries specified by the LI command has been exceeded. You may specify 10 libraries.
L-ML BDT	Multiple block data subprogram. Attempted to initialize the same area more than once.
L-ML EMA	Illegal EMA declaration.
L-NO IDS	Not enough ID segments to finish the load.
L-NO RSG	No root segment specified.
L-NO SNP	Notify your System Manager. The loader could not find the file \$SYENT.
L-OV BASE	Base page overflow.
L-OV DSK	Program exceeds the maximum disk space allowed a program.
L-OV FIX	This is a fixup table overflow.
L-OV MEM	The relocation address has exceeded 77777B, 77777B-MSEG, or the size specified using SZ,N.
L-OV PTN	This size specified using the SZ command was too large for the program's type.
L-OV RBP	Overflow of rotating base page (MLLDR only).

L-OV SAV	Overflowed SAVE area.
L-OV SNP	Overflowed snap file \$SYENT. Try running the loader using RU, LOADR, -1, -1 to create the \$SYENT file.
L-OV SYM	This is a symbol table overflow.
L-PE LDR	Tried to do a purge, replace, or permanent load with a copy of the loader.
L-RE SEQ	Record out of sequence.
L-RF EMA	Attempt to access an EMA external with offset or indirect.
L-RP CPY	Attempt to replace a copied program.
L-RP MLS	Tried to use MLLDR to replace a program that was loaded by LOADR or vice-versa. Another cause is trying to replace or purge a memory-resident program using MLLDR.
L-RP PGM	Tried to replace or purge a permanent program that has terminated serially reusable, saving resources, or was operator suspended.
L-RQ PGS	Both a SZ and AS were used and the size is larger than the partition.
L-SH EMA	A shareable EMA label file did not have the control command \$SHEMA starting in the first column in the first line, or there was another error in the file.
L-SH PTN	A program cannot be assigned to a shareable EMA partition, a subpartition of a shareable EMA partition, or a mother partition that has any shareable EMA subpartitions.
L-SS ENT	Attempt to access an SSGA entry point without asking for SSGA at the beginning of the load. Reload the program but this time do an OP,SS at the beginning of the load.
L-SZ ALC	Allocate size error.
L-SZ EMA	EMA size is greater than 1K pages and VMA is not being used.
L-TR ADD	No transfer address. Only subroutines were loaded.
L-UN EXT	Undefined externals exist that prohibit the load from completing.
L-VS EMA	Tried to use shareable VMA (not supported).
L-VS EMA	The specified VMA size is illegal.
W-DU PGM	Duplicate program name.
W-IL CMD	Attempted to relocate a module or transfer to a command file while doing special processing when undefined externals exist.

W-IN CAP	Due to insufficient user capability, the program priority was changed to 99. The message is: /MLLDR: XXXXX SET TO PRIORITY 99 /MLLDR: W-IN CAP
W-RQ PGS	The required number of pages is too large.
W-SV MIX	Mixing SAVE named COMMON with named COMMON.
W-UN EXT	Undefined externals exist and the loader was initiated from an interactive device.
W-WS EMA	The specified working set size is too large.

Logon Error Codes

LGON 00	Session environment not initialized.
LGON 01	FMP error on account file access.
LGON 03	Session limit exceeded.
LGON 04	No such user.
LGON 05	Illegal access.
LGON 06	Conflict in definition of session LU.
LGON 07	No room for Session Control Block (SCB).
LGON 08	Duplicate session identifier.
LGON 09	SST overflow.
LGON 10	No free ID segments or FMGR not found.
LGON 11	FMP error on disk mount attempt.
LGON 12	Account file corrupt.
LGON 13	Conflict with system disk LU.
LGON 14	Bad job logon request
LGON 15	Session Primary Program not found.
LGON 16	Number of UDSPs or depth invalid.

LU Lock Error Codes

LU01	A program has one or more LUs locked and is trying to lock another with wait.
LU02	Illegal LU reference.
LU03	Not enough parameters in the call; LU reference is less than one; or LU not locked to caller.
LU04	Trying to lock an LU not defined in caller's SST.

Outspool Error Messages

Message	Explanation
JOB WAIT ON PT	End of tape occurred between :JO and :EO commands.
JOB WAIT ON SPOOL RESOURCE	Required spool file or logical device cannot be obtained at this time.
JOB WAIT ON EXTENT	Spool file overflows available disk space.
END JOB ABNORM	JOBFIL could not be opened; or other uncorrectable error occurred; or JOB was run before spool initialization.
BAD EOF	Message appears after last line of file ASCII file outspooling overflowed; or was otherwise incomplete.

Parity Error Messages

Hard Parity Error Message (reproducible).

PE PG# nnnnn BAD	(physical page # of parity error)
ABE aaaaaa bbbbbb e	(A, B, and E-Registers)
XYO xxxxxx yyyyyy o	(X, Y, and O-Registers)
PE ppppp mmmmmm ppppp ABORTED	(program name and logical memory address of parity error)

Soft Parity Error Message (not reproducible):

SOFT PE PG# nnnnn	(physical page# of parity error)
ABE aaaaaa bbbbbb e	(A, B, and E-Registers)
XYO xxxxxx yyyyyy o	(X, Y, and O-Registers)
PE ppppp mmmmmm ppppp ABORTED	(program name and logical memory address of parity error)

Pascal Errors

Program Errors

Message Number	Error Message
1	Undefined cas in line xxxx.
2	Heap/Stack collision in line xxxx.
3	Nil pointer dereferenced in line xxxx.
4	Value out of range in line xxxx.
5	MOD by invalid value in line xxxx.
6	String underflow in line xxxx.
7	String overflow in line xxxx.
8	String bad index in line xxxx.
9	Invalid string in line xxxx.
91	Dispose called with a nil ptr in line xxxx.
92	Disposed of an invalid variant in line xxxx.
93	Release called with a nil ptr in line xxxx.
94	Dispose called with a bad ptr in line xxxx.
95	Release called with a bad ptr in line xxxx.
96	Overflow of two-word integer in line xxxx.
97	Illegal char for base in line xxxx.
98	No value to convert in line xxxx.
99	Insufficient image space in line xxxx.

Pascal I/O Errors

Message Number	Error Message
1	Unexpected EOF
2	File must be text
3	File must be direct
4	Bad record length
5	Must reset or open file
6	Must rewrite or open file
7	Direct access read error
8	Sequential access read error
9	Invalid real number read
10	Line read was too long
11	Invalid real number read
12	File is not CCTL
13	No scratch file available
14	Neg FLD/DEC field width not allowed
15	File cannot be type 1 or 2
16	File must be type 1 or 2
17	Cannot open LU 0 for read only
18	Missing file name
19	File is not open
20	Identifier not in enumerated type
21	Value not in enumerated type

READT/WRITT Error Codes

READ 001	The requested mag tape unit is down.
READ 002	The mag tape READT is trying to restore contains information in a format not restorable by READT.
READ 003	The mag tape unit you wish to use is locked to a process.
READ 004	The parameter describing the desired mag tape unit does not satisfy READT's requirements for a legal mag tape LU.
READ 005	The desired mag tape unit is offline.
READ 006	READT rejected the use of the specified disk LU.
READ 007	The driver detected a parity error when reading from the mag tape.
READ 008	The end of tape was reached.
READ 009	The desired cartridge has a file open or the cartridge is locked to another program.
READ 010	You are operating in a non-session environment. An LU must be specified (negative LU) because there is not a free disk pool.
READ 011	READT rejected the size (number of tracks) you specified.
READ 012	The routine READT uses to mount a cartridge detected an error.
READ 013	The desired disk LU or the available free LUs in the disk pool are not large enough to restore the cartridge that is on the mag tape.
READ 014	The FMP tracks on LU 2 or LU 3 (if 3 exists) are not restorable with READT.
READ 015	The private/group attribute of the cartridge tape file conflicts with the attribute of the mounted cartridge. Run READT with the correct P/G option, or mount a compatible cartridge.
READ 016	Bad transmission – mag tape to memory rec xxx READT detected an error in transmission of data from the mag tape unit into memory. Try reading the tape again. If it happens once more call your system manager.
READ 017	Internal buffer too small – the tape records are longer than the READT internal buffer. The buffer size must be increased and READT must be reloaded.

READ 018	READT aborted by user. This message is produced when you respond NO to any prompt, or when READT is halted using the BReak command.
READ 019	Disk error on LU xx, TRACK xxx – READT encountered an error when reading the listed track of the listed LU.
READ 020	Verify error on track xxxx – a compare error was encountered when verifying the listed track.
READ 021	Invalid parameter – An invalid parameter was specified in the READT command runstring. Check the runstring and re-enter the parameter.
WRIT 001	The requested tape LU is not active.
WRIT 002	Only the system manager can save system disks.
WRIT 003	The mag tape you wish to use is locked to a process.
WRIT 004	The parameter describing the desired mag tape unit does not satisfy READT's requirements for a legal mag tape unit.
WRIT 005	The desired mag tape unit is offline.
WRIT 006	A write ring is required to write information on a mag tape.
WRIT 007	The driver detected a parity error when reading from the mag tape.
WRIT 008	The end-of-tape was reached.
WRIT 009	The desired cartridge has a file open or the cartridge is locked to another program.
WRIT 010	The desired cartridge or disk LU could not be found.
WRIT 011	WRITT rejected the use of the specified disk LU.
WRIT 012	You cannot save FMP tracks off LU 2 or LU 3 with WRITT.
WRIT 013	WRITT tried to read data from a disk LU into memory and found the transmission irregular. Run WRITT again; if the situation occurs once more there may be a bad track on that disk LU. Save as much data as you can and notify your system manager.
WRIT 014	The transmission of data from memory to mag tape may be faulty. Run WRITT again, if it happens once more call your system manager.
WRIT 016	An error was detected in transmission of data from the magnetic tape to memory. If this error recurs, the tape may be faulty; see the system manager.

Reconfiguration Error Codes

Configuration Error	Meaning
1	Invalid LU number or a bit bucket LU.
2	Illegal select code number.
3	New select code entered is identical to new select code assigned to disk, system console or list device, or else the current select code entered is identical to the old select code for disk, system console or list device (i.e., do not reconfigure that which was already done via the SWITCH register).
10	Specified total number of pages outside the range.
11	Invalid bad page number.
12	Specified SAM extension entry beyond physical memory size due to bad pages.
13	Current running total exceeds available pages in block of good memory or exceeds size of mother partition.
14	Second parameter of partition definition entry other than RT, BG or S, or else S was entered when a subpartition definition was not expected.
15	Third parameter of partition definition entry other than R.
16	No such program, or the name of a segment was entered or invalid type was entered for partition assignment.
17	Invalid partition number.
18	Program does not fit in the assigned partition.
19	Invalid number of pages was entered for program size.
20	Number of defined partitions already equal to allowed maximum number and more undefined pages remain.
21	Page requirements of an EMA program cannot be modified.
22	Number of pages in SAM extension requires division into more than five blocks.

23	An illegal label was entered.
24	Too many shareable EMA partitions.
25	An undefined label was entered during the shareable EMA program phase of reconfiguration.
26	Shareable EMA partition specified is too small.
27	Program assigned to a shareable EMA partition.
28	Assigning more than one label to the same partition.
29	A duplicate label was entered.
30	Tried to assign a program to a shareable EMA partition when no shareable EMA partitions have been defined.
31	No ID extensions left.

Resource Number Errors

RN00	There are no option bits set in the call.
RN01	No resource numbers in system.
RN02	The specified resource number is not defined.
RN03	An unauthorized attempt was made to clear a local resource number.

Schedule Call Error Codes

SC00	A batch program attempted to suspend (EXEC(7)).
SC01	Missing parameter.
SC02	Illegal parameter.
SC03	The specified program cannot be scheduled.
SC04	The specified program is not a subordinate (or “son”) to the program issuing the completion call.
SC05	The program given is not defined. The format for the SC05 error is: PROG/SEGMENT <i>nnnnn</i> SC05 <i>name</i> ADDRESS

where:

nnnnn is the program/segment not found
name is the calling program name.

SC06	No resolution code is specified in the execution time EXEC call (not 1, 2, 3, or 4).
SC07	A prohibited memory lock was attempted.
SC08	The program just scheduled is assigned to a partition smaller than the program itself or to an undefined partition.
SC09	The program just scheduled is too large for any partition of the same type.
SC10	There is not enough System Available Memory (SAM) for the string passage.
SC11	EXEC schedule or timed execution request was issued and program specified is already in the time list for another session.
SC12	The program tried to do an EXEC 8 call to load an MLS program.
SC13	The main program and segments were not SP'ed onto the same disk cartridge.
SC14	Track ownership in Track Assignment Table (TAT) does not correspond to ID segment for program's segment.

SMP Error Messages

Error Message	Meaning
SMP:LU <i>xx</i> EOFER <i>filename</i>	File <i>filename</i> just outspooled to LU <i>xx</i> overflowed or was otherwise incomplete.
SMP:LU <i>xx</i> DOWN <i>filename</i> HELD	LU <i>xx</i> down: <i>filename</i> placed in active hold.
SMP:FMP - <i>nn</i> <i>filename</i> ::crn	FMP error - <i>nn</i> occurred during SMP operation. Usually indicates loss of JOBFIL or SPLCON.
SMP:File deleted from SPLCON!	The file would not be correctly opened and was cleared from SPLCON.

System and Break-Mode Command Error Messages

Error Message	Meaning
OP CODE ERROR NO SUCH PROG	Illegal operator request code. The name entered is not a main program in the system.
INPUT ERROR	A parameter is illegal.
ILLEGAL STATUS	Program is already scheduled.
CMD IGNORED-NO MEM	Not enough System Available Memory exists for storing the program's command string.
ILLEGAL PART'N	Partition does not match command request.
SIZE ERROR	0 Program larger than requested size (RQ)
SIZE ERROR	1 MSEG too small
SIZE ERROR	2 MSEG too large
SIZE ERROR	3 Larger than largest partition
SIZE ERROR	4 Attempted to change MSEG of SHEMA program
xxxxx NO SWAP TRACKS	Not enough swap tracks available to swap out a program on behalf of program xxxxx.

Track (Disk Parity) Error Message

TR nnnnn EQT xx,Uyy S
U

nnnnn	Track number of track containing error.
xx	EQT of disk.
yy	Subchannel of disk.
S	System request encountered error.
U	User request encountered error.

VMA/EMA Error Codes

VMA/EMA errors that cause a program to abort have the same format as the MP and DM error returns:

VM *xx*
or
EM *xx*

where:

<i>xx</i>	Is an FMP error number (if <i>xx</i> is less than 80). FMP reports the error as a negative number and VMA/EMA reports the same error as the last two digits of the positive of that number. VMA/EMA errors with <i>xx</i> greater than 80 are errors from the VMA/EMA routines.
01	Disk error.
02	Duplicate file name.
05	Backing store file created with less than 256 blocks of memory or file extent cannot be created when read only access has been specified to the VMA file. X-Register = the requested page ID that caused the problem.
06	File not found or directory read protected.
07	Illegal security code or illegal write on LU 2 or 3 or directory write protected.
08	File open or cartridge containing file is locked.
09	No such directory (FMP-209).
12	File extent cannot be created when read only access has been specified to the VMA file. X-Register = requested page that caused the problem.
13	Specified cartridge is locked.
14	Directory full.
15	Illegal file name.
19	Illegal access on a system disk.
20	An array is specified with incorrect subscripts.
21	MSEG in the \$EMA or \$VMA directive is not specified correctly.
22	The program is not an VMA/EMA program.
32	Cartridge not found.
33	Not enough room on cartridge.
40	LU not found in SST.

- 46 Greater than 255 file extents on the VMA file.
- 80 VMA/EMA system is corrupt.
- 81 Not an VMA/EMA program, or a bad request to VMAIO, EMAIO, or XLUEX.
- 82 Requested page beyond maximum page specified for VMA/EMA system or the VMA disk file is too small.
X-Register = requested page number (in octal).
Abort address = address of instruction causing program to abort.
- 83 All pages locked; working set is not large enough to support the size of MSEG specified in your program.
- 84 The backing store file is not the correct type (file type 2) or the record length is not 1024 words.
- 85 VMA scratch file cannot be purged; file is in use by another program.
- 86 Access to VMA system after the VMA file has been closed.
- 87 MSEG is too small.
- 88 Cannot re-specify the VMA file.
- 89 Transfer too big for VMAIO, EMAIO, or XLUEX.
- 90 Shareable EMA size for program is larger than the shareable EMA area already allocated.
- 91 Program and shareable VMA/EMA area are assigned to the same partition or program is assigned to a reserved partition in which the program's shareable VMA/EMA area has already been allocated.