HP 3000 Computer Systems

MPE IV
software pocket guide

HEWLETT PACKARD

19447 Pruneridge Ave., Cupertino, Ca. 95014

Part No. 30000-90049

Printed In U.S.A.
Jan 1981

NOTICE

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

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS 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 program language without the prior written consent of Hewlett-Packard Company.

Copyright © 1981 by HEWLETT-PACKARD COMPANY

LIST OF EFFECTIVE PAGES

Seventh Edition . . . . . Jan 1981
Update No. 1 . . . . . Apr 1981

Changed Pages **Effective Date**

iii to iv . . . . . Apr 1981
viii . . . . . Apr 1981
1-9 to 1-10 . . . . . Apr 1981
2-6 to 2-6a. . . . . Apr 1981
8-6 to 8-9 . . . . . Apr 1981
8-12 to 8-14. . . . . Apr 1981
10-7 to 10-15. . . . . Apr 1981

PRINTING HISTORY

Seventh Edition . . . . . Jan 1981
Update No. 1 . . . . . Apr 1981

CONTENTS

Section I
COMMANDS

: to :ALTACCT 1-1
:ALTGROUP 1-2
:ALTLOG to :BASIC 1-3
:BASICGO Tto :BUILD 1-4
:BYE to :DEALLOCATE 1-5
:DEBUG to :DSTAT 1-6
:EDITOR to :FILE 1-7
:FORTGO to :FREERIN 1-11
:GETLOG to :HELLO 1-12
:HELP to :IML 1-13
:JOB to :JOBPRI 1-14
:LISTACCT to :LISTF 1-15
:LISTLOG to :LISTVS 1-18
:MPLINE to :NEWGROUP 1-19
:NEWUSER to :PREP 1-20
:PREPRUN 1-21
:PTAPE to :RELEASE 1-22
:RELLOG to :REPORT 1-23
:RESET to :RESTORE 1-24
:RESUME to :RPGPREP 1-25
:RUN to :SECURE 1-26
:SEGMENTER to :SETCATALOG 1-26a
:SETDUMP to :SHOWJCW 1-27
:SHOWJOB to :SHOWOUT 1-28
:SHOWQ to :SPLGO 1-29
:SPLPREP to :SYSDUMP 1-30
:TELL to :VSUSER 1-31
Compiler Subsystem Commands 1-32
Parameterlist Options 1-32
Commands 1-33
Default Capabilities 1-35

Section II
CONSOLE COMMANDS

:ABORTIO/=ABORTIO to :DOWN 2-1
:DOWNLOAD to :IMLCONTROL 2-2
:JOBFENCE to =LOGON 2-2a
:MPLINE to :STARTSPOOL 2-3
:STOPSPOOL to :VMOUNT 2-3a

CONTENTS (Continued)

:WARN to :WELCOME 2-4
Control and Maintenance Processor (CMP) 2-4
CMP Commands 2-4a
Spooling Command/Event Matrix 2-5
System Start-Up 2-6
Stand-Alone Memory Dump 2-7
DPAN (Dump Analyzer) 2-9

Section III
EDIT,FCOPY, SORT, MERGE

EDITOR
Operation 3-1
Commands 3-1

FCOPY
Operation 3-5
Specifying Carriage Control 3-5
Deblocking Records 3-5
Copying Multiple Tape Files 3-5
Translating Code 3-5
Omitting User Labels 3-5
Selecting Subsets of Records 3-5a
Shifting 3-6
Skipping EOF 3-6
Ignoring errors 3-6
Creating new file 3-6
Verifying Copy 3-6
Comparing from file with to file 3-6
Displaying numerical codes 3-6
Displaying characters 3-6
Determine sequence for
copying KSAM file 3-6
Copy data from KSAM file 3-6
Examples 3-7

SORT
Operation 3-8
Commands 3-8
Intrinsics 3-10

MERGE
Operation 3-11
Commands 3-11
Intrinsics 3-12

Section IV
IMAGE, QUERY

IMAGE
SCHEMA Processor
Operation 4-1
File Designators 4-1
Commands. 4-1
SCHEMA Structure 4-2
DBLOAD. 4-3
DBRECOV. 4-3
DBRESTOR. 4-4
DBSTORE. 4-4
DBUNLOAD 4-4
DBUTIL 4-5
Calling an IMAGE procedure. 4-7
Intrinsics Exceptional Conditions. 4-10
QUERY
Operation 4-12
Commands. 4-12
Statements. 4-16
Statement Parameters 4-17

Section V
KSAM, V/3000

KSAMUTIL
Operation 5-1
Commands. 5-1
SPL Intrinsics. 5-3
COBOL Procedures 5-10
BASIC Procedures. 5-12
FORTRAN Procedures. 5-14
V/3000
FORMSPEC. 5-15
REFSPEC 5-22
REFORMAT 5-24
COMAREA 5-27
RPG Interface. 5-29
ENTRY. 5-32

CONTENTS (Continued)

Section VI
UTILITIES

ASOCTABL to DISKED2 6-1
DPAN4 6-3
FREE2 to LISTDIR2 6-5
LISTLOG2 to MEMTIMER 6-6
MEMLOGAN 6-7
PATCH to SADUTIL 6-8
SLPATCH to SPOOK 6-11
VINIT 6-13

Section VII
SEGMENTER

Operation 7-1
Commands 7-1
Intrinsics 7-2

Section VIII
INTRINSICS

ACTIVATE to CLEANUSL 8-1
CLOCK to CREATEPROCESS 8-2
CTRANSLATE to DMOVIN 8-3
DMOVOUT to FCLOSE 8-4
FCONTROL 8-5
FDELETE to FFILEINFO 8-6
FGETINFO to FLOCK 8-8
FMTCALENDAR to FREADBACKWARD 8-9
FREADDIR to FSETMODE 8-12
FSPACE to GENMESSAGE 8-13
GET to GETPROCINFO 8-14
GETUSERMODE to LOCKGLORIN 8-15
LOCKLOCRIN to PCONTROL 8-16
POPEN to PRINTFILEINFO 8-6a
PRINTOP to QUIT 8-17
QUITPROG to SETDUMP 8-18
SETJCW to UNLOCKLOCRIN 8-19
WHO to XCONTRAP 8-20
XLIBTRAP to ZSIZE 8-21

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Section IX
DEBUG

Operation 9-1
Access Scope 9-1
Messages 9-1
Command Syntax 9-2
Command Operation 9-2
Breadpoint Commands 9-3
Display/Listing Commands 9-5a
Memory/Register Modifaction Commands 9-8
Calculation Display Command 9-9
Trace Command 9-10
Segment Freeze Commands 9-10
Segment and Register Contents 9-11
Stack Marker Format 9-12
PMAP Format 9-12
LMAP Format 9-14
Condition Codes 9-15
Status Register 9-15

Section X
FILE SYSTEM

Input Set 10-1
Output Set 10-1
File Codes 10-2
Carriage-Control Directives 10-3
Carriage-Control Effect Summary 10-5
File Access/Security 10-6
Account, Group, and File Default Security 10-6
Net Default Access 10-6
Run Time Errors 10-7
File System Errors 10-7

Section XI
ASCII, INSTRUCTION SET

ASCII Character Set 11-1
Index of OP Code Groups 11-3
Series II/III Instruction Set 11-5
Series 30/33/44 Instruction Set 11-15

CONTENTS (Continued)

Section XII
SPECIAL KEYS AND CODES

Special Terminal Keys 12-1
029 Card-Punch Transliterations 12-1
ASCII Character Substitutes 12-1

CONVENTIONS USED IN THIS MANUAL

Notation	Description
[ ]	An element inside brackets is optional. Several elements stacked inside a pair of brackets indicates the user may select any one or none of these elements. Example: [A] User may select [B] A or B or neither
{ }	An element inside braces is required. Several elements stacked inside a pair of braces indicates the user must select one of these elements. Example: {A} User must select {B} A or B
Bold type	Bold types indicates required elements. It may be an operation, a command, an intrinsic, a required parameter, or the required portion of a command. Example: **:RUN FREE2.PUB.SYS** (operation) **:BREAKJOB** (command) **:TEXT** (required portion of a command) **FCHECK (filenum. . .)** (intrinsic and required parameter)
Lowercase parameter	User-supplied variable. The parameter must be replaced by a user-supplied variable. Example: ;PASS = password ;**PASS** = XYZ
return	A carriage return.

CONVENTIONS (Continued)

Conventions which apply to intrinsics only.

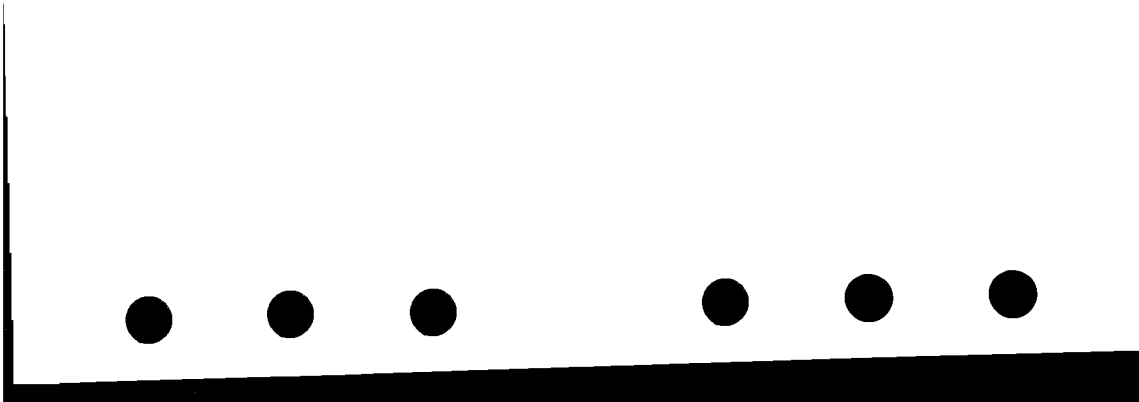
Superscripts

O-V	Option variable
B	Byte
I	Integer
L	Logical
D	Double
A	Array
V	By value (no superscript indicates by reference)

Symbols

=	"means"
|	"or"
:=	"is assigned"
%	octal

Commands



:([:] commandname) [sessionname,] username [/userpass]
.acctname [/acctpass]
[,groupname[/grouppass]]

:ABORT

:ALLOCATE [PROGRAM, / PROCEDURE,] name

Default: Program

Capability: OP

:ALTACCT **acctname**
[;PASS = [password]]
[;FILES = [filespace]]
[;CPU = [cpu]]
[;CONNECT = [connect]]
[;CAP = [capabilitylist]]
[;ACCESS = [fileaccess]]
[;MAXPRI = [subqueuename]]
[;LOCATTR = [localattribute]]
[;VS = [volset: { ALT / SPAN }]]

Defaults: Unlimited filespace, unlimited cpu time, unlimited connect time, AM, AL, GL, SF, ND, IA, BA capabilities, no security restrictions at the account level, CS subqueue, double-word 0 localattribute.

Capability: SM

Note: If acctname is SYS, and the fileaccess parameter is omitted, the default security is R, X: ANY; A, W, L:AC.

filespace = disc storage limit (sectors)

Capabilitylist = SM (System Manager)
AM (Account Manager)
AL (Account Librarian)
GL (Group Librarian)
DI (Diagnostician)
OP (System Supervisor)
SF (Save files – perm.)
ND (Non-sharable Device)
CS (Use Communications Subsystems)
UV (Use Volumes)
CV (Create Volumes)
PH (Process Handling)
DS (Extra Data Segments)
MR (Multiple RIN'S)
PM (Privileged Mode)
IA (Interactive Access)
BA (Local Batch Access)

Default is AM, AL, GL, SF, ND, IA, and BA

fileaccess = (modelist:userlist,. . .)
See File Access Security.

Default: R, A, W, L, X: AC

subqueuename = BS, CS, DS, ES

localattribute = (Defined by installation)

When entire keyword group is omitted, parameter remains unchanged for account.

:ALTGROUP **groupname**
[;PASS=[password]] [;CAP=[capabilitylist]]
[;FILES=[filespace]] [;CPU= [cpu]]
[;CONNECT=[connect]] [;ACCESS=[fileaccess]]
[;VS=[volset[: {ALT | SPAN}]]]

Defaults: IA, BA capabilities, unlimited filespace, unlimited cpu time, R, A, W, L, X, S: GU (all groups except PUB); or R, X: ANY; A, W, L, S: AL, GU (PUB group only).

Capability: AM

:ALTLOG **logid** [;LOG=logfile{,DISC / ,TAPE}]
[;PASS=password]

:ALTSEC **filereference** [;{[modelist :userlist[;. . .]]}]

Note: For modelist and userlist options, see File Access Security.

:ALTUSER **username**
[;PASS=[password]] [;CAP=[capabilitylist]]
[;MAXPRI=[subqueuename]]
[;LOCATTR=[localattribute]]
[;HOME=[homegroupname]]

Defaults: SF, ND, IA, BA capabilities, CS subqueue, double-word 0 localattribute, no homegroup.

Capability: AM

Note: For parameter definitions and options, see SYSTEM CAPABILITY SETS.

:ALTVSET **vsname**
{;ADDCLASS=vcname:vname[,vname
[,. . .[,vname]]]
;EXPANDCLASS=vcname:vname[,vname
[,. . .[,vname]]]
;EXPANDSET=vname:type[[,. . .[,vname:type]]]}

Capability: SM or AM with CV

:APL

:ASSOCIATE devclass

:BASIC [commandfile] [,[inputfile] [,listfile]]

Default: $STDINX, $STDLIST

Note: Formal file designators – BASCOM, BASIN, BASLIST

Commands

:BASICGO [commandfile] [,listfile]

Default: $STDINX and $STDLIST

Note: Formal file designators – BSCTEXT, BSCLIST

:BASICOMP [commandfile] [,[uslfile][,listfile]]

Default: $STDINX, $NEWPASS, $STDLIST

Note: Formal file designators – BSCTEXT, BSCUSL, BSCLIST

For a discussion of BASIC compiler parameters, see pg 1-32.

:BASICPREP [commandfile] [,[progfile] [,listfile]]

Default: $STDINX, $NEWPASS, $STDLIST

Note: Formal file designators –BSCTEXT, BSCPROG, BSCLIST

:BUILD **filereference**

```
[;DEV=[dsdevice]#] [device] ]
[;DISC=[numrec] [,[numextents]
        [,initalloc] ] ]
[;REC=[recsize] [,[blockfactor]
[ , [ F ] [ ,BINARY ] ] ] ]
[   [ U ] [ ,ASCII  ] ] ]
[   [ V ] [         ] ] ]
[;CCTL   ]
[;NOCCTL ]
[;TEMP]
[;CODE=[filecode] ]
[;RIO   ]
[;NORIO ]
[;MSG   ]
[;CIR   ]
```

Default: Record size: determined at configuration
Blockfactor: never less than 1
File length: F for disc files, U for all others
Code: BINARY
Carriage Control: NOCCTL
TEMP: Permanent file

Relative input/output: NORIO
File type: STD

Note: For recsize, positive value indicates words and negative value indicates bytes.

:BYE

:COBOL [textfile] [,[uslfile]
[,[listfile]
[,[masterfile] [,newfile]]]]

Default: $STDIN, $NEWPASS, $STDLIST
For a discussion of COBOL compiler parameters, see page 1-32.

:COBOLGO [textfile] [,[listfile]
[,[masterfile] [, newfile]]]

Default: $STDIN, $STDLIST, input read from textfile, no new file written.

:COBOLPREP [textfile] [,[progfile]
[,[listfile] [,[masterfile]
[,newfile]]]]

Default: $STDIN, $NEWPASS, $STDLIST

:COMMENT [text]

Default: Null comment

:CONTINUE

:DATA [jsname,] username[/userpass]
.acctname[/acctpass] [;filename]

Default: No job/session name. No distinguishing filename assignment.

:DEALLOCATE [PROGRAM, / PROCEDURE,] **name**

Default: Program file

Capability: OP

Commands

:DEBUG

Capability: PM

:DISASSOCIATE devclass

:DISMOUNT [* | vcsname][.groupname[.acctname]]

Default: *

=DSLINE { { ,OPEN | ,SHUT } [, [MASTER | SLAVE] [,speed]] | ldn { ,TRACE { ,ON[,[ALL][,[mask] [,[numentries]] [,[WRAP][,filename]]]] | ,OFF } } [,COMP | ,NOCOMP] }

:DSLINE To open a hardwired line:

dsdevice[;LINEBUF=buffsize] [;EXCLUSIVE]

To open a telephone line:

dsdevice[;LINEBUF=buffsize]

[;LOCID=local-id-sequence]
[;REMID=remote-id-sequence1[,remote-id-sequence2]....]
[;PHNUM=telephone number]
[;EXCLUSIVE]
[;COMP]
[;NOCOMP]
[;QUIET]

To close a communication line:

dsdevice
[ds-line-number] [;CLOSE]
@

:DSTAT [ldn | ALL]

Default: Only the status of non-system discs displayed

:EDITOR [listfile]

Default: $STDLIST. If specified with no device parameter, default device is LP.

Note: See EDIT Subsystem.
Formal file designator: EDTLIST

:ELSE

:ENDIF

:EOD

:EOF:

Note: The last colon in this command must be followed by a blank.

:EOJ

:FCOPY [FROM=[{Filename / * / <empty>}];TO=[{(dfile,kfile) / filename / * / <empty>}]

[;parm list]]

Using :FCOPY without parameters accesses FCOPY subsystem. For a full discussion of parm list, see pp.3-5,6,7.

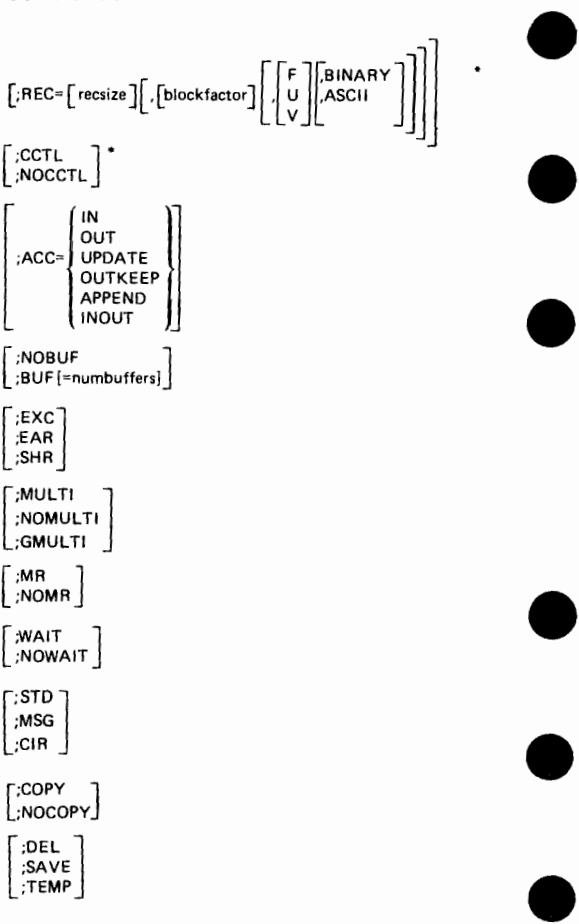
:FILE

Note: Parameters used with the :FILE command depend on the type of file described.

New or Old (Existing) Files

:FILE formaldesignator [=$NEWPASS / [=filereference] [,NEW] / =$OLDPASS / [=filereference] [,OLD / ,OLDTEMP]]

Commands



*Indicates not used for old disc files.

```
[ ;NOLABEL                                                  ]
[ ;LABEL=[volid] [,[type] [,[expdate][,seq]]]]]]
[;FORMS=formsmsg]

[ ;NOLOCK ]
[ ;LOCK   ]

[;DEV=[[dsdevise]#] [device] [,[outpri] [,numcopies] ] ]
[;CODE=[filecode] ]
[;DISC=[numrec] [,[numextents] [,initalloc] ] ]
[;ENV=environmentfilename]
[;DEN= [1600]]
[      [6250]]
```

Defaults: for new files are:
recsize = 128 for DISC/TAPE
= -132 for LP
= -80 for CARD/PUNCH
= -72 for TERM
blockfactor = physical record/recsize
SHR (if ACC= IN, otherwise EXC)
BUF = 2
DEV = DISC
CODE = 0
DISC = 1023,8,1

Note: The parameter group [;DISC=[numrec] [,[numextents] [,initalloc]]] cannot be included if the parameter group

```
                [,OLD      ]
[=filereference] [,OLDTEMP ] is specified.
```

For recsize, positive value indicates words and negative value indicates bytes.

For outputpriority, select a number between 1 (lowest priority) and 13 (highest priority) (Default is 1.)

For filecode see File System

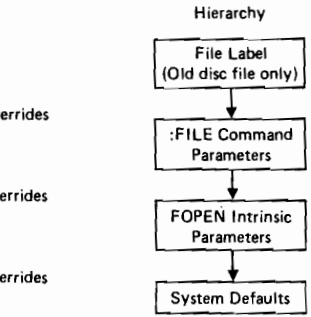
Commands

User Pre-Defined (Back-referenced) Files

```
:FILE formaldesignator = *formaldesignator
```

For System-Defined Files:

```
:FILE          formaldesignator = $NULL

:FILE          formaldesignator = { $STDIN  }
                                  { $STDINX }
                                  { $STDLIST }

[;REC=[recsize] [ ,[blockfactor] [ [ F ] ,BINARY ] ] ]
                                 [ [ U ] ,ASCII  ]
                                 [ [ V ]         ]

[ ;CCTL   ]
[ ;NOCCTL ]

[;DEV=[[dsdevice]#] [device] [,[outpri] [,numcopies] ] ]

[ ;ACC= { IN      } ]
        { OUT     }
        { UPDATE  }
        { OUTKEEP }
        { APPEND  }
        { INOUT   }

[ ;NOBUF          ]
[ ;BUF [=numbuffers] ]

[ ;EXC ]
[ ;EAR ]
[ ;SHR ]

[ ;MR   ]
[ ;NOMR ]

[ ;WAIT   ]
[ ;NOWAIT ]

[ ;NOLABEL                                                    ]
[ ;LABEL=[volid] [,[type] [,[exdate] [,seq] ] ]               ]

[;FORMS=formsmsg]

[ ;NOLOCK ]
[ ;LOCK   ]
```



:FORTGO [textfile] [,[listfile] [,[masterfile] [,[newfile]]]]

Default: $STDIN, $STDLIST

Note: Formal file designators FTNTEXT, FTNLIST, FTNMAST, FTNNEW

:FORTPREP [textfile] [,[progfile] [,[listfile] [,[masterfile] [,newfile]]]]

Defaults: $STDIN, $NEWPASS, $STDLIST

Note: Formal file designators FTNTEXT, FTNLIST, FTNMAST, FTNNEW, FTNPROC

:FORTRAN [textfile] [,[uslfile] [,[listfile] [,[masterfile] [,newfile]]]]

Defaults: $STDIN, $NEWPASS, $STDLIST

Note: Formal file designators, FTNTEXT, FTNLIST, FTNMAST, FTNNEW, FTNUSL

For a discussion of FORTRAN compiler parameters, see page 1-32.

:FREERIN rin

:GETLOG logid; LOG=logfile[,DISC / ,TAPE]
[;PASS=password]

:GETRIN **rinpassword**

:HELLO [sessionname,] **username**[/userpass]
.**acctname** [/acctpass] [,groupname
[/grouppass]]
[;TERM=termtype]
[;TIME=cpusecs]
[;PRI = BS / CS / DS / ES]
[;INPRI=inputpriority / ;HIPRI]

Note: The termtype parameter determines the type of terminal used, as follows:

0 = ASR 33 EIA-compatible HP 2749B (10 characters per second (cps)).

1 = ASR 37 Teleprinter (10 cps).

2 = ASR 35 EIA-compatible (10 cps).

3 = Execuport 300 Data Communications Transceiver Terminal (10, 15, 20 cps).

4 = HP 2600A or DATAPOINT 3300 (10-240 cps).

5 = Memorex 1240 (10, 15, 30, 60 cps).

6 = HP 2762A/B (GE Terminet 300 or 1200) or Data Communication Terminal, Model B (10, 15, 30 120 cps).

9 = HP 2615A (MiniBee) (10-240 cps).

10 = HP 2640A/B, HP 2641A, HP 2644A or HP 2645A Character mode or program control of block mode (10-240 cps).

11 = HP 2640A/B, HP 2641A, HP 2644A or HP 2645A. Character mode and block mode without program control. (Not block/page mode.)

12 = HP 2645A Katakana/Roman Terminal.

13 = Message Switching Network or Other Computer.

14 = Multi-Point Terminal.

15 = HP 2635A Printing Terminal.
8-bit protocol (for second character set).

16 = HP 2635A Printing Terminal.
7-bit protocol (for standard character set).

Terminal types 4, 6, 9, 10, 11, 12, 15 and 16 only are available on Series 30/33 systems.

If a user at a CRT operating at 2400 baud specifies termtype = 10, and the CRT is not a 2640 terminal, the terminal will cease to accept data. To correct, enter F^C, log off, and log on again with the correct term type.

For cpusecs enter maximum cpu time allowed from 1 to 32767 seconds or "?" or "UNLIM" for no limit.

For subqueue, enter BS, CS, DS or ES. (Users and accounts must have valid MAXPRI to access requested queue.) (Default: CS)

:HELP

```
[ HELP
  tablecontents
  command [,keyword]
          [,ALL    ]
  udcname
  ALL              ]
```

Note: :HELP udcname is not recognized when used inside a subsystem.

:IF

[{} logexpr[}] THEN

Note: See also :ELSE and :ENDIF

:IML

```
[ ENHANCE= { 0 }          ] [;BLANKS]
           { 1 }
           { 2 }
           { 3 }          ]
```

Note: The ENHANCE parameter alters the display as follows:

Option	3270 Normal Intensity	3270 High Intensity
0	264x half bright	264x normal
1	264x normal	264x underline
2	264x normal	264x inverse video
3	264x inverse video	264x normal

:JOB [jobname,] **username** [/userpass]
.acctname [/acctpass] [,groupname
[/grouppass]]
[;TIME=cpusecs]
[;PRI = { BS | CS | DS | ES }]

[;HIPRI | ;INPRI=inputpriority]

[;RESTART]
[;OUTCLASS=[device] [,outputpriority]
[,numcopies]]

Default: $STDLIST,Priority=DS

Note: For other parameter meanings, see :HELLO

:JOBPRI [0 | maxqueue] [,defaultqueue]

Capability = OP

Note: 0 = No maxqueue limit.
maxqueue, defaultqueue = CS, DS, or ES

Defaults: No change in maximum priority, no change in execution priority.

:LISTACCT [acctset] [,listfile]

Defaults: All accounts (System Manager capability required), $STDLIST

Capability: AM or SM

:LISTF [fileset][, 0 / 1 / 2 / -1] [;listfile]

Note: 0
Listfilename.

1
List 0 plus file code, record size, type ASCII/binary. CCTL, EOF pointer position, and maximum records allowed.

2
List 1, plus blocking factor, sectors used, extents allocated, extents allowed.

-1
Octal listing of file label.

(0, 1, 2 may be requested for any permanent file by anyone. -1 lists file label and requires SM/AM capability; SM lists any file, and AM lists any file in acct.

The characters @, #, and ? can be used as wild card characters in the fileset parameter. These wild card characters have the following meanings:

@ – specifies zero or more alphanumeric characters.

– specifies one numeric character.

? – specifies one alphanumeric character.

The characters can be used as follows:

n@	List all files starting with the character *n*.
@*n*	List all files ending with the character *n*.
n@*x*	List all files starting with the character *n* and ending with the character *x*.

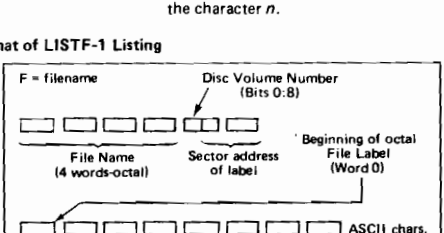
n##..# List all files starting with the character *n* followed by up to seven digits (useful for listing all EDIT/3000 temporary files).

?*n*@ List all files whose second character is *n*.

n? List all two-character files starting with the character *n*.

?*n* List all two-character files ending with the character *n*.

Format of LISTF-1 Listing



The disc file label contains the following:

Words		Contents	Words (Octal)
0-3		Local file name.	0-3
4-7		Group name.	4-7
8-11		Account name.	10-13
12-15		Identity of file creator.	14-17
16-19		File lockword.	20-23
20-21		File security matrix.	24-25
22	(Bits 0:15)	Not used.	26
	(Bit 15:1)	File secure bit: If 1, file secured. If 0, file released.	
23		File creation date*	27
24		Last access date.*	30
25		Last modification date.*	31
26		File code.	32
27		File control block vector.	33

*Dates in same format as return value for CALENDAR intrinsic.

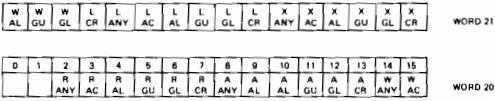
Format of LISTF-1 Listing (continued)

Words		Contents	Words (Octal)
28	(Bit 0:1)	Store Bit. (If on, :STORE or :RESTORE, in progress.)	34
	(Bit 1:1)	Restore Bit. (If on, :RESTORE in progress.)	
	(Bit 2:1)	Load Bit. (If on, program file is loaded.)	
	(Bit 3:1)	Exclusive Bit. (If on, file is opened with exclusive access.)	
	(Bits 4:4)	Device sub-type.	
	(Bits 8:6)	Device type.	
	(Bit 14:1)	File is open for write.	
	(Bit 15:1)	File is open for read.	
29	(Bits 0:8)	Number of user labels written.	35
	(Bits 8:8)	Number of user labels.	
30-31		Maximum number of logical records.	36-37
32		Unused	40
33		Private Volume Information	41
34		Checksum	42
35		Cold-load identity.	43
36		Foptions specifications.	44
37		Logical record size (in negative bytes).	45
38		Block size (in words).	46
39	(Bits 0:8)	Sector offset to data.	47
	(Bits 8:3)	Not used.	
	(Bits 11:5)	Number of extents minus 1.	
40		Logical size of last block.	50
41		Extent size.	51
42043		Number of logical records in file.	52-53
44-107		Two-word addresses of up to 32 disc extents, beginning with address of first extent (words 44–45).	54-133
108-109		Restore time	134-135
110		Restore date	136
124-127		Device class	154-157

File Name

Filename may consist of from one to eight alphanumeric characters. Must begin with a letter; special characters not allowed.

File Security Matrix



R	READ	ANY:	ANY USER
A	APPEND	AC:	Account Member
W	WRITE	AL:	Account Librarian
L	LOCK	GU:	Group User
X	EXECUTE	GL:	Group Librarian
		CR:	Creator

:LISTLOG [logid[;PASS]]

:LISTGROUP [@ / group / groupset] [,listfile]

Default: All groups in the log-on account.

Capability: AM or SM

:LISTUSER [@ / user] [,listfile]

Default: All users in the log-on account.

Capability: AM or SM

:LISTVS [vslist] [,0 / ,1 / ,2] [;listfile]

```
                 { OPEN [,filename]                                  }
                 { UP, upentry                                       }
                 { DOWN, downentry                                   }
                 { SHUT [,NOW]                                       }
:MPLINE ldn      { MESSAGES [,ON ]                                   }
                 {          [,OFF]                                   }
                 {                                                   }
                 {          { ,ON[,[ALL] [,[mask] [,[numentries]]    }
                 { TRACE    {      [,[WRAP] [,filename] ] ] ] ]      }
                 {          { ,OFF                                   }
```

Capability: Console Operator

```
:MOUNT          [   *     ] [.groupname[.acctname] ]
                 vcsname
                           [;GEN=[genindex] ]
```

Default: *

:MRJE

```
:NEWACCT        acctname, mgrname
                [;PASS=[password] ]
                [;FILES= [filespace] ]
                [;CPU=[cpu] ]
                [;CONNECT=[connect] ]
                [;CAP=[capabilitylist] ]
                [;ACCESS=[fileaccess] ]
                [;MAXPRI=[subqueuename] ]
                [;LOCATTR=[localattribute] ]
                [;VS=volset:SPAN]
```

Defaults: No password assigned, unlimited filespace, unlimited cpu, unlimited connect AM, AL, GL, SF, ND, IA, BA capabilities, R, A, L, W, X: AC fileaccess CS subqueue.

Capability: SM

Note: For parameter definitions and options, see System Capability Sets

```
:NEWGROUP       groupname
                [;PASS=[password] ]
                [;CAP=[capabilitylist] ]
                [;FILES=[filespace] ]
                [;CPU=[cpu] ]
                [;CONNECT=[connect] ]
                [;ACCESS=[fileaccess] ]
                [;VS=[volset[:SPAN] ] ]
```

Defaults: No password is assigned, IA, BA capabilities, filespace equals account's, cpu equals account's, connect equals account's, R, X: ANY, A, W, L, S: AL, GU for PUB group and R, A, W, L, X, S: GU for all other groups.

Capability: AM

:NEWUSER **username**
[;PASS=[password]]
[;CAP=[capabilitylist]]
[;MAXPRI=[subqueuename]]
[;LOCATTR=[localattribute]]
[;HOME=[homegroupname]]

Defaults: No password is assigned, SF, ND, IA, BA, CS subqueue.

Capability: AM

Note: For parameter definitions and options, see System Capability Sets.

:NEWVSET **vsname**
;MEMBERS=vname:type[,vname:type,
. . .[vname:type]
[;CLASS=vcname:vname[,vname,. . .[,vname]]

Capability: CV

:PREP **uslfile,progfile**
[;ZERODB]
[;PMAP]
[;MAXDATA=segsize]
[;STACK=stacksize]
[;DL=dlsize]
[;CAP=caplist]
[;RL=filename]
[;PATCH=patchsize]

Default: No PMAP listing, MPE assumes segsize will change, stacksize is estimated by MPE segmenter, dlsize is estimated by MPE segmenter.

Note: segsize
Max DL to Z size, in words.

stacksize
Initial Q to Z area, in words.

dlsize
Initial DL to DB area, in words

caplist

IA	Interactive access	standard capabilities
BA	Local batch access	
PH	Process handling	
DS	Extra Data Segment management	
MR	Multiple resource management	
PM	Privileged-mode operation	

If no "CAP"= is specified, both IA and BA are assigned. If only IA or BA is requested, only that access is assigned.

Formal file designator is SEGLIST

:**PREPRUN** **uslfile**[,entrypoint]
[;NOPRIV]
[;PMAP]
[;DEBUG]
[;LMAP]
[;ZERODB]
[;MAXDATA=segsize]
[;PARM=parameternum]
[;STACK=stacksize]
[;DL=dlsize]
[;LIB={G / P / S}]
[;CAP=caplist]
[;RL=filename]
[;NOCB]

Default: Primary entry point, segments of privileged mode program will remain in privileged mode, no PMAP listing, no breakpoint is set, no **LMAP** listing, MPE assumes segsize will not change, parameternum is Q (initial) 0-4 address is filled with zeros, MPE segmenter estimates stacksize and dlsize, System Library.

Note: Formal file designator for PMAP is SEGLIST and for LMAP is LOADLIST.

Note: P – Account Public Library
G – Group Library
S – System Library (Default)

For other parameters, see :PREP.

:PTAPE filename

:PURGE **filereference**[,TEMP]

Default: Permanent file is assumed

:PURGEACCT **acctname**[;VS=volset]

Capability: SM

:PURGEGROUP **groupname**[;VS=volset]

Capability: AM

:PURGEUSER **username**

Capability: AM

:PURGEVSET vsname

Capability: SM, or AM with CV

:RECALL

:REDO

:RELEASE **filereference**

:RELLOG **logid**

:REMOTE [dslinenumber] [mpecommand]

:REMOTE HELLO

[sessionname,] **username**[/userpass] **.acctname** [/acctpass]

[,groupname[/grouppass]]
[;TERM=termtype]
[;TIME=cpusecs]
[;PRI={BS / CS / DS / ES}]
[;INPRI=inputpriority / ;HIPRI]
[;DSLINE=dsdevice]

Default: No session name assigned, no cpusecs limit assigned, CS priority class, inputpriority of 8, HIPRI is current and execution limit.

:RENAME **oldfilereference,**
newfilereference[,TEMP]

Default: Permanent file

Note: Both newfilereference and oldfilereference have the format:filename [/lockword] [.groupname] [.acctname] The command does not change the file domain. Use SAVE to make TEMP file permanent.

:REPORT [groupset] [,listfile] [;VS=volset]

Defaults: User – his/her own group, AM – all groups in his/her own account, SM – all groups in all accounts.

Capability: AM or SM.

Note: *groupset* Specifies the accounts and groups for which information is to be listed. The permissible entries and the capability required (shown in parentheses) are as follows: Account Manager is shown as AM; System Manager as SM.

groupdesig Reports on the specified group in the log-on account. Standard user can only specify his log-on group.

@ Reports on all groups in the log-on account (AM or SM).

groupdesig.acctdesig Reports on the specified group in the account (SM).

@.*acctdesig* Reports on all groups in the specified accounts (SM).

@,@ Reports on all groups in all accounts (SM).

Default: For standard user: his own group

For Account Manager: All groups in his own account.

For System Manager: All groups in all accounts.

:RESET { @ | formaldesignator }

:RESETACCT [@ | acctname] [,CPU | ,CONNECT]

Capability: SM

:RESETDUMP

:RESTORE **restorefile** [[file [.group[.account]] | ; [@ [.group[.account]] | .@ [.@ | .account]]] [,...]] [[;KEEP] | [;DEV=device] | [;SHOW] | [;FILES=maxfiles] | [;OLDATE]]

Default: Users – all files in log-on group, AM – all files in all groups in log-on account, SM – all files in system.

Capability: See System Manager/System Supervisor Reference Manual, :RESTORE command, "Operation".

Note: :RESTORE requires SF access to group or SM/OP capability. (AM has SF access to all groups in his account). Formal file designator is SYSLIST.

:RESUME

:RESUMELOG

:RJE [commandfile] [,[inputfile] [,[listfile] [,punchfile]]]

Default: $STDIN, $STDLIST, $NEWPASS

Note: Formal file designator, RJECOM, RJEIN, RJEPUNCH, RJELIST

:RPG [textfile] [,[uslfile] [,[listfile] [,[masterfile] [,newfile]]]]

Default: $STDIN, $NEWPASS, $STDLIST

Note: Formal file designators RPGTEXT, RPGUSL, RPGLIST, RPGMAST, RPGNEW.

For a discussion of RPG compiler parameters, see page 1-32.

:RPGGO [textfile] [,[listfile] [,[masterfile] [,newfile]]]

Default: $STDIN, $STDLIST, newfile is no file written.

Note: Formal file designators, RPGTEXT, RPGLIST, RPGMAST, RPGNEW.

:RPGPREP [textfile] [,[progfile] [,[listfile] [,[masterfile] [,newfile]]]]

Default: $STDIN, $NEWPASS, $STDLIST.

Note:Formal file designators, RPGTEXT, RPGPROG, RPGLIST, RPGMAST, RPGNEW.

:RUN **progfile**[,entrypoint]
[;NOPRIV]
[;LMAP]
[;DEBUG]
[;MAXDATA=segsize]
[;PARM=parameternum]
[;STACK=stacksize]
[;DL=dlsize]
[;LIB = {P / G / S}]
[;NOCB]
[;INFO=string]
[;STDIN = [*formaldesig / fileref / $NULL]]
[;STDLIST = [*formaldesig / fileref[,NEW] / $NULL]]

Default: Primary entry point, privileged mode programs remain in privileged mode, no LMAP listing, DEBUG call is not issued, MPE assumes segsize will not change, parameternum is Q (initial) -4 address is filled with zeros, stacksize is estimated by segmenter, dlsize is estimated by segmenter, System Library.

Note: Certain parameter meanings are described under :PREP, and :PREPRUN.
Formal file designator LMAP is LOADLIST.

:SAVE { $OLDPASS, newfilereference / tempfilereference }

Note: newfilereference is required as a parameter if $OLDPASS is used. tempfilereference is required as a parameter if $OLDPASS/newfilereference is not used.

:SECURE **filereference**

:SEGMENTER [listfile]

Default: $STDLIST

Note: Formal file designator, SEGLIST
See also Segmenter section.

:SETCATALOG [catfilename[,catfilename,. . .,
[catfilename]]] [;SHOW] [;ACCOUNT / ;SYSTEM]

Capability: ACCOUNT parameter requires AM, SYSTEM parameter requires SM

:SETDUMP [{ DB[,ST] [,QS] / ST[,DB] / QS[,DB] } [;ASCII]]

Defaults: DB – settings of all registers at time of abort and stack marker trace ASCII – octal.

:SETJCW jcwname char value

:SETMSG { ON / OFF }

:SHOWALLOW [{ username / @ } . { acctname / @ }]

:SHOWCATALOG [listfile]

Default: $STDLIST

:SHOWDEV [ldev / classname]

Default: Status information for all devices is displayed.

:SHOWIN [#Innn / STATUS]

[SP] [;item] [;item] [;item]

Default: Information for all input devicefiles displayed.

Note: items: [DEV=ldev]

[JOB = { @J / @S / #Jnnn / #Snnn }]

[ACTIVE / READY / OPENED]

Do not use duplicate item keywords in this command

:SHOWJCW [jcwname]

Default: Status of all JCW's displayed

Commands

:SHOWJOB [[#]Snnn | [#]Jnnn | STATUS | id[;state] | state[;id]]

Defaults: Status information for all jobs/sessions displayed.

Note: id: @ | @J | @S | [jsname,]username.acctname | @,username.acctname | [@,]@.acctname

state: INTRO | WAIT[,N | ,D] | EXEC | SUSP

N = Non-deferred
D = Deferred

:SHOWLOG

Capability: OP

:SHOWLOGSTATUS logid

:SHOWME

:SHOWOUT [#Onnn | STATUS]

[SP] [;item] [;item] [;item]

Default: Status information for all output devicefiles displayed.

Note: items: [DEV={ ldev / classname }]
[JOB= @J / @S / @ / [#]Jnnn / [#]Snnn]
[ACTIVE / READY [,N / ,D] / OPENED / LOCKED]

Do not use duplicate item keywords in this command.

:SHOWQ

Capability: OP

:SHOWTIME

=SHUTDOWN

:SPEED newinspeed, newoutspeed

Note: Valid values for inspeed and outspeed are: 10, 14, 15, 30, 60, 120, 240 (480 and 960 for Series 30/33 only). These characters represent characters-per-second for terminal I/O.

:SPL [textfile] [,[uslfile] [,[listfile] [,[masterfile] [,newfile]]]]

Defaults: $STDIN, $NEWPASS, $STDLIST

Note: Formal file designators, SPLTEXT, SPLUSL, SPLLIST, SPLMAST, SPLNEW.

For a discussion of SPL compiler parameters, see page 1-32.

:SPLGO [textfile] [,[listfile] [,[masterfile] [,newfile]]]

Defaults: $STDIN, $STDLIST

Note: Formal file designators, SPLTEXT, SPLLIST, SPLMAST, SPLNEW

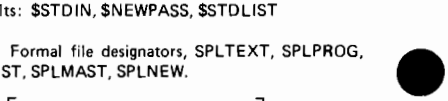
Commands

:SPLPREP [textfile] [,[progfile]
[,[listfile] [,[masterfile]
[,newfile]]]]

Defaults: $STDIN, $NEWPASS, $STDLIST

Note: Formal file designators, SPLTEXT, SPLPROG, SPLLIST, SPLMAST, SPLNEW.

:STORE



Default: All files in log-on group.

Capability: Users with SM or OP capability can store any user file in the system. Users with AM capability can store any file in the account (but cannot dump those with negative file codes unless they have PM capability also.)

Note: System Manager has read access to all files. Account Manager has read access to all files in his/her account.

The :LISTF command also applies to the :STORE command. Formal file designator is SYSLIST.

:STREAM [inputfile] [,character]

Default character for : prompt replacement is !

:SWITCHLOG

Capability: OP

:SYSDUMP **dumpfile** [,auxlistfile]

Default: $STDLIST.

Capability: SM or OP

Note: The formal file designator used by the :SYSDUMP command executor for this file is SYSDLIST; the formal file designator used by the MPE segmenter is SEGLIST.

:TELL { [jsname,] username.acctname | [#] Jnnn | [#] Snnn | @ | @.acctname | @J | @S } [;] [text]

:TELLOP [text]

:TUNE [MINCLOCKCYCLE] [;CQ | ;DQ | ;EQ [=[BASE] [,[LIMIT] [,[MIN] [,MAX]]]]]

Note: More than one of CQ, DQ and EQ may be specified in a TUNE command. Separate phrases with a semicolon.

:VINIT [listdevice]

Default: $STDLIST

Capability: SM or OP

:VSUSER [vsname]

Commands

COMPILER SUBSYSTEM COMMANDS

For BASICOMP, COBOL, FORTRAN, RPG, SPL,

$CONTROL parameterlist

Parameterlist Options
(Separated by Commas)

For ALL Compilers
[[NO] LIST] [[NO] SOURCE] [[NO] WARN]
[[NO] MAP] [[NO] CODE]
[LINES=nn] [USLINIT]

Additional for SPL
[SEGMENT=segname] [ADR] [MAIN=pgname]
[INNERLIST] [UNCALLABLE]
[SUBPROGRAM[=(proc(*) . . .)]]
[PRIVILEGED] [ERRORS=nn]

Additional for FORTRAN
[INIT] [BOUNDS] [FILE=nn [-nn]]
[FIXED/FREE] [[NO] LABEL] [SEGMENT=
name] [ERRORS=nn] [CHECK=nn]
[CROSSREF [ALL]] [[NO] LOCATION]
[[NO] STAT] [[NO] CODE]
[[NO] LIST] [USLINIT] [[NO] MAP]
[[NO] SOURCE] [[NO] WARN]

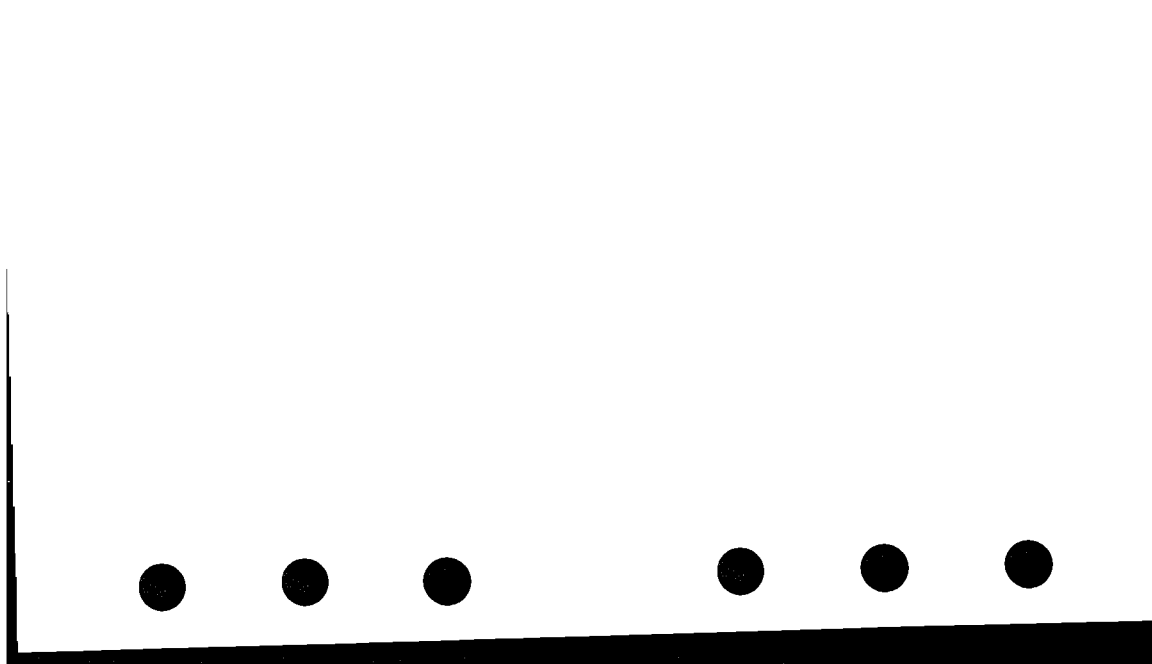
Additional for COBOL
[QUOTE=$\left\{ \begin{matrix} ' \\ " \end{matrix} \right\}$] [DYNAMIC] [SUBPROGRAM]
[DEBUG] [BOUNDS] [ERRORS=nn]

Additional for COBOLII
[ANSISUB] [BOUNDS] [CHECKSYNTAX]
[[NO] CODE] [[NO] CROSSREF] [DEBUG]
[DYNAMIC] [ERRORS=n] [LINES=pagenum]
[[NO] LIST] [LOCKING] $\left[\begin{matrix} \text{LOCON} \\ \text{LOCOFF} \end{matrix} \right]$
[[NO] MAP] [[NO] MIXED] [QUOTE=]
[[NO] SOURCE] $\left[\begin{matrix} \text{STDWARN=level} \\ \text{NOSTDWARN} \end{matrix} \right]$
[SUBPROGRAM] [USLINIT] [[NO] VERBS]
[[NO] WARN]

Parameterlist Options (continued)

Additional for RPG
[QUOTE={ ' / " }] [SEG=n] [ERRORS=nn]

Additional for BASIC Compiler
[START=programname] [SUBPROGRAM] [INIT]
[[NO] LABEL] [SEGMENT=segname]



For All Compilers

$TITLE [[string] [,string] . . .]

For COBOL, FORTRAN, RPG, SPL only

$EDIT [VOID=sequencenumber]

[,SEQNUM=sequencenumber]
[,NOSEQ]

[,INC=incnumber]

$IF [$X_n$ = {OFF / ON}]

$PAGE [string[,string] . . .]

$SET [$X_n$ = {OFF / ON} [, X_n {OFF / ON}] . . .]

For FORTRAN, SPL only

$TRACE [programunit] ; identifier [,identifier] [, . . .]

For BASICOMP only

$COMPILE progname [(entrypoint$_1$, . . . entrypoint$_n$)]
. . . [,progname$_n$[(entrypoint$_1$ . . . , entrypoint$_n$)]]

$ENTRY progname$_1$ . . . [,progname$_n$]

$EXIT

Note: To transmit a command to *newfile*, precede it by an *additional* $.

Commands

	Mnemonic	Meaning
User/Acct	SM	(System Manager)
	AM	(Account Manager)
	AL	(Account Librarian)
	GL	(Group Librarian)
	DI	(Diagnostician)
	OP	(System Supervisor)
	SF	(Save files = perm.)
	ND	(Non-sharable Device)
	CS	(Use Communications Subsystems)
	UV	(Use Volumes)
	CV	(Create Volumes)
Group	PH	(Process Handling)
	DS	(Extra Data Segments)
	MR	(Multiple RINs)
	PM	(Privileged Mode)
	IA	(Interactive Access)
	BA	(Local Batch Access)

DEFAULT CAPABILITIES

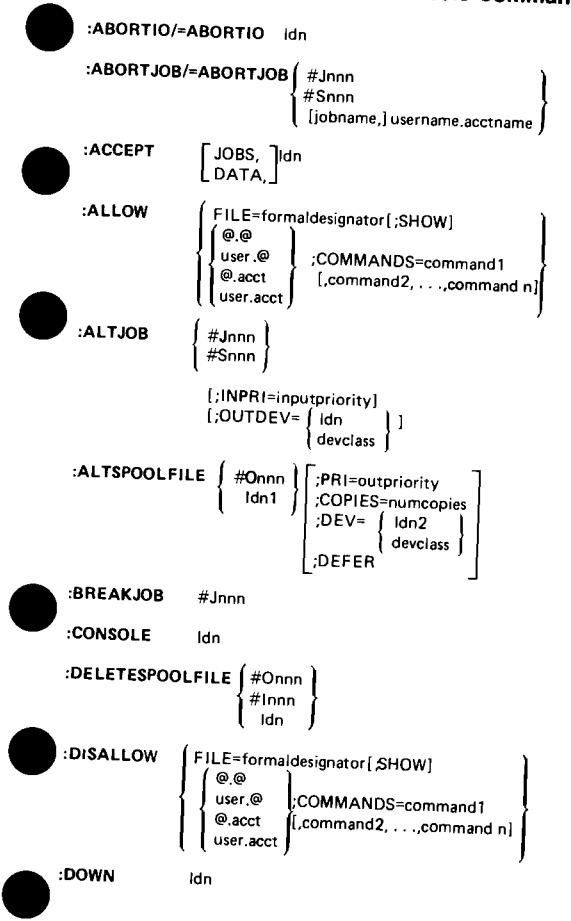
Accounts	AM, AL, GL, SF, ND, IA, BA
Groups	IA, BA
Users	SF, ND, IA, BA

CAPABILITY CHECKING

Time	Checking
Log-on	User ⩽ Account
:NEWGROUP :ALTGROUP	Group ⩽ Account
:NEWUSER :ALTUSER	User ⩽ Account
:PREP	File ⩽ User
:RUN	File ⩽ Group
Intrinsic Call	Program file capability

Section II

Console Commands


:ABORTIO/=ABORTIO ldn
:ABORTJOB/=ABORTJOB #Jnnn
#Snnn
[jobname,] username.acctname
:ACCEPT JOBS,
DATA,
ldn
:ALLOW FILE=formaldesignator[;SHOW]
@.@
user.@
@.acct
user.acct
;COMMANDS=command1
[,command2, . . .,command n]
:ALTJOB #Jnnn
#Snnn
[;INPRI=inputpriority]
[;OUTDEV= ldn
devclass]
:ALTSPOOLFILE #Onnn
ldn1
;PRI=outpriority
;COPIES=numcopies
;DEV= ldn2
devclass
;DEFER
:BREAKJOB #Jnnn
:CONSOLE ldn
:DELETESPOOLFILE #Onnn
#Innn
ldn
:DISALLOW FILE=formaldesignator[;SHOW]
@.@
user.@
@.acct
user.acct
;COMMANDS=command1
[,command2, . . .,command n]
:DOWN ldn

Console Commands

```
:DOWNLOAD   ldn [,filename
                ,MARGIN=nn]

:DSCONTROL  {ldn      } [;OPEN] [, [MASTER] [,[SPEED=]]
            {devclass} [;SHUT]   [SLAVE]   speed]]

                      [          {,ON[,[ALL] [,[mask]]
                      [;TRACE    {     [,[numentries]
                      [          {     [,[WRAP]
                      [          {     [,filename]]]]]
                      [          {,OFF                }]

                      [;COMP]
                      [;NOCOMP]

                      [;MON [,DS]]
                      [     [,CS]]
                      [;MOFF      ]

:FOREIGN    ldn

:GIVE       ldn

:HEADOFF    ldn

:HEADON     ldn

:IMLCONTROL {START configfilename [;TRACEON [,[ALL]
            {                     [,[mask]
            {                     [,[numentries]
            {                     [,[WRAP]
            {                     [,Filename]]]]]]
            {STOP configfilename
            {KILL configfilename
            {TRACE configfilename ON [,[ALL] [,[mask]
            {                        [,[numentries]
            {                        [,[WRAP]
            {                        [,filename]]]]]
            {TRACE configfilename OFF
```

:JOBFENCE priorityfence

Note: $0 \leq$ priorityfence ≤ 14 (large is more limiting)

:JOBSECURITY { HIGH / LOW }

:LDISMOUNT vcsname.group.account

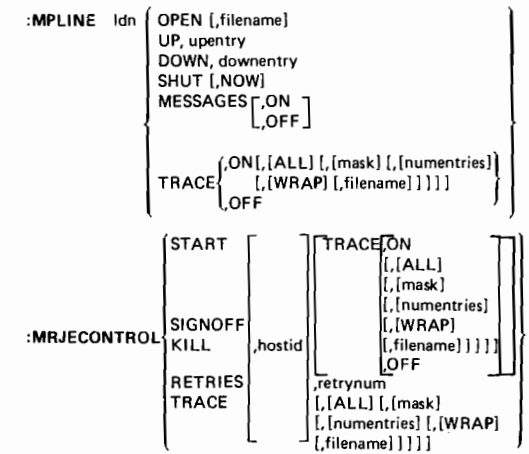
:LIMIT { [numberjobs] [,numbersessions] }

:LMOUNT vcsname.group.account[;GEN=genindex]

:LOG logid, { START / RESTART / STOP }

=LOGOFF

=LOGON


:MPLINE ldn
OPEN [,filename]
UP, upentry
DOWN, downentry
SHUT [,NOW]
MESSAGES [,ON / ,OFF]
TRACE ,ON[,[ALL] [,[mask] [,[numentries] [,[WRAP] [,filename]]]]] / ,OFF
:MRJECONTROL
START ,hostid
SIGNOFF ,hostid
KILL ,hostid
RETRIES ,retrynum
TRACE ,hostid [,[ALL] [,[mask] [,[numentries] [,[WRAP] [,filename]]]]]
TRACE [,ON / ,OFF]
[,[ALL]
[,[mask]
[,[numentries]
[,[WRAP]
[,filename]]]]]
:OUTFENCE outputpriority[;LDEV=ldn]
Note: 1 ≤ outputpriority ≤ 14 (large is more limiting)
:REFUSE [JOB, / DATA,] ldn
:REPLY/=REPLY pin,reply[,param]
:RESUMEJOB #Jnnn
:RESUMESPOOL ldn [;BACK / ;FORWARD / ;BEGINNING] [nnn FILES / nnn PAGES]
:SHOWCOM ldn[;ERRORS] [;RESET]
=SHUTDOWN
:STARTSPOOL ldn[;SHUTQ] / devclass

:STOPSPOOL { ldn[;OPENQ] | devclass }

:STREAMS { ldn | OFF }

:SUSPENDSPOOL ldn[;FINISH]

:TAKE ldn

:UP ldn

:VMOUNT { ON [,AUTO] | OFF } [;ALL]

Console Commands

:WARN { [#] Jnnn
[#] Snnn
[jsname,] username.acctname
@
[@.] acctname
@J
@S } [;] [text]

:WELCOME return

#message return
#message return
#return

For Series 44:

Control and Maintenance Processor (CMP)

When the system is running, the CMP will usually be inactive, and commands are entered via MPE in the usual manner. However, if for any reason you are unable to communicate with MPE, enter a CNTL B from the system console; the operating system (MPE) is suspended, the CMP is enabled, and you are automatically prompted for an MPE-like command. If the operating system is not functioning, as in the case of a system halt, access the CMP simply by entering a carriage return.

CMP Commands:

DISPLAY

DUMP

HALT

HELP

IOMAP

LOAD

LOG

RUN

SELFTEST

SHUTTEST

SPEED

START

Spooling Command/Event Matrix

System Start Up

Type	From	Thumb-wheel	Sw Reg	Effect:
WARM*	DISC	WARM	%0004 **	Permits recovery of incompletely processed spooled jobs and spoolfiles.
COOL	DISC	WARM	%0004 **	Standard operation. All permanent user files are saved. All temporary files, jobs, and sessions done before COOLSTART are lost.
COLD	SERIAL STORAGE DEVICE	COLD	%3006	System files and I/O configuration come from tape; user files, directory, accounting information, and global RINs are retained -- obtained from disc.
UPDATE	SERIAL STORAGE DEVICE	COLD	%3006	System files come from tape; I/O configuration, directory, accounting information, and global RINs remain unchanged.
RELOAD	SERIAL STORAGE DEVICE	COLD	%3006	Complete MPE comes from tape -- system files, I/O configuration, file directory, and user files. If the disc label is good, some items are not reloaded. To insure that the entire system is restored, follow the RELOAD with an update option.

* All cold-load options except WARM allow the operator to alter the input/output device configuration currently in effect.

** Value should be the DRT number of the system disc.

Note: Thumbwheels apply to Series 30/33/44; Switch Registers apply to Series II/III.

Note: If your system has an HP-IB Interface Module, the low-order bits (rightmost eight bits) of the System Switch Register must always be set to %175 for System Startup. The high-order bits (leftmost eight bits) are set to the octal representation of the DRT number of the cold load device. For additional information, consult the Console Operator's Guide, part number 32002-90004.

Stand-Alone Memory Dump

For Series II/III

1. Mount tape with write-ring on Tape Unit 0. Bring to load point and set on-line.
2. If computer running, press RUN/HALT.
3. While pressing (and holding) the ENABLE switch, press the DUMP switch; a preconfigured pattern will appear in the SYSTEM SWITCH REGISTER.
4. If the computer halts with the correct number for your memory size appearing in the CURRENT INSTRUCTION REGISTER, the dump is complete. If the computer does not halt with the correct address, mount another tape and go to step 3.

Note: The dump is done off-line, and necessitates subsequent system start-up to resume system operation. It is a diagnostic tool which should not be confused with the :SYSDUMP command described on page 26.

Proper Current Instruction Register (CIR) Contents After Dump

Memory Size (Words)	CIR Contents (Octal)
128K	000002
160K	100002
192K	000003
224K	100003
256K	000004
384K	000006
512K	000010
768K	000014
1024K	000020

Console Commands

For Series 30/33

Operator Function: Dumping Main Memory

1. Mount a serial disc or magnetic tape on a logical device specified by the device class DDUMP, then place the drive on-line.

2. On the System Front Panel, set the DUMP (Series 30) or MEMORY DUMP (Series 33) thumbwheel switch to the octal value of the DRT number (channel address and device address) of the system disc.

3. From the System Front Panel, press the DUMP (Series 30) or MEMORY DUMP (Series 33) key.

   OR

   From the System Console, press the DUMP key while pressing the CNTL key. (Be sure the Front Panel keys are enabled.)

4. At this point, SDF will attempt to serially execute a file (SDFCOM) containing ASCII formatted commands (which are located on the system disc) until a WARMSTART command is encountered.

   The Software Dump Facility is loaded and the following message appears on the console:

   SOFTWARE DUMP FACILITY (VER XX.XX/XX)

   and the system halts.

   Insert or mount a serial storage device, place the drive on-line and press the RUN key. This will initiate the dump to the device previously specified in the device class DDUMP. Main memory is stored to the serial storage device, and the system pauses awaiting further instructions. At this point you may start up the system using any of the system startup options.

   NOTE: Entering a CNTL Y from the console causes SDF to abort the current command and read all remaining commands from the console.

For Series 44

Operator Function: Dumping Main Memory

1. Mount a serial disc or magnetic tape on a logical device specified by the device class DDUMP, then place the drive on-line.

2. On the System Control Panel, set the DUMP thumbwheel switch to the octal value of the DRT number (channel address and device address) of the system disc drive.

3. From the System Control Panel, press the DUMP key.

OR

From the CMP, enter DUMP at the prompt.

4. At this point, SDF will attempt to serially execute a file (SDFCOM) containing ASCII formatted commands (which are located on the system disc) until one of the following are encountered; a WARMSTART command, a HALT command, or an end-of-file condition.

   If the Software Dump Facility is loaded correctly the following message will appear on the console:

   SOFTWARE DUMP FACILITY (VER XX.XX/XX)

   The system will then halt.

5. When the HALT light comes on, the console operator should check to see that the serial storage device is on-line and ready, then press the RUN key on the System Control Panel, or enter RUN in response to the CMP prompt. Main memory is stored to the serial storage device, and the system pauses awaiting further instructions. At this point you may start up the system using any of the system startup options.

   NOTE: Entering a CNTL Y from the console causes SDF to abort the current command and read all remaining commands from the console.

DPAN4 (DUMP ANALYZER)

To Invoke:

:RUN DPAN4[.groupname.acctname] [;PARM=10]

Note: Entering PARM=10 initiates the interactive dialogue between you and DPAN4. For more information, consult the MPE System Utilities Reference Manual (Part No. 30000-90044).

Respond with tape number to message:

?!O/time/#{J/S}jsnum/pin/LDEV#FOR "MDUMP"
ON TAPE (NUM).

DPAN4 output is transmitted to $STDLIST unless run from session; then output is to DEV=LP.

If you print a dump on a system which is different from the one on which the dump was taken, make certain that you have a copy of the original (dump system) LOADMAP file. Enter a :FILE command:

```
:FILE LOADMAP.PUB.SYS=
      dumploadmap.grp.acct
```

then run DPAN4.

Section III

EDIT, FCOPY, SORT, MERGE

Operation

:EDITOR

or

:FILE name;**DEV=** {ldev / devclass}

:EDITOR *name

or

:FILE EDTTEXT=name

:RUN EDITOR.PUB.SYS, BASICENTRY

Commands

Note: In these commands, "IN", "TO", and "BY" can be replaced by commas",".

/**ADD**[**Q**] [linenumber] [,HOLD[Q] [,NOW]]

/**BEGIN**[**Q**]

/**CHANGE**[**Q**] {col [/col] / string} TO string [IN rangelist]

/**COPY**[Q] range TO linenumber [BY increment]

/**DELETE**[**Q**] [rangelist]

/{**END** / **EXIT**}

/**FIND**[**Q**] range

Note: Second position in range is upper bound.

/**GATHER**[**Q**] {range TO linenumber / ALL[TO linenumber]} [BY increment]

/**HOLD**[**Q**][range [,APPEND]]

Note: Null range implies clear hold file.

/**INSERT**[**Q**] position[BY increment] [,HOLD[Q] [,NOW]]

EDIT

/JOIN[Q] filename

[({ flinenumber[/flinenumber] })]
[({ #filerecnum[/#filercnum] })]

[TO linenumber]

[BY increment] [[,] UNNUMBERED]

Note: flinenumber and #filerecnum applies to filename. Linenumber applies to workfile.

/{KEEP[filename] [(range)] [,UNNUMBERED]
 KEEPQ filename }

Note: Q-option does not allow (range) and UNNUMBERED.

/LIST[Q] [range] [,UNNUMBERED] [,OFFLINE]
[,TRANSLATE] [,NOTEXT]

/MODIFY[Q] [rangelist]

/NOT

/OR

/PROCEDURE [procedurename, [S P G] [,rangelist]]

/Q [string "Z::"]

/REPLACE[Q] [rangelist] [,HOLD[Q] [,NOW]]

/SET optionlist

Note: Options are separated by commas.

options: FROM=linenumber DELTA=increment
DEPTH=integer LENGTH=colnum
LEFT=colnum RIGHT=colnum
TIME[S]=integer SIZE=integer
LINES=maxlines TABCHAR[=string]

[LONG / SHORT] [REAR / FRONT] [DISPLAY / QUIET] [FIXED / VARIABLE]

[POLL / BATCH] [FORMAT={DEFAULT / COBOL}]

[TABS[=(colnum[,colnum]...)] / NOTABS]

/TEXT [file[((linenum/linenum / #recnum/#recnum))] [,UNNUMBERED]]

/USE [file]

/VERIFY optionlist

Note: optionlist includes all SET command keywords, plus TOTAL, FILES, and ALL.

/WHILE (FLAG)

Note: To find all locations of string in a text file and allow user to modify lines in textfile, enter:

```
WHILE
  FIND "string"
  MODIFY*
```

/XPLAIN [command / ALL] [,OFFLINE]

/YES

/Z ::=

Note: Use to set up a command parameter string. Then use Z:: embedded within a command to invoke this string.

/: MPE command

Note: MPE commands may be passed in this way only if they are programmatically executable.

Notes:

1. Example linenumber : 1, 20, 30.5,9.999, FIRST,LAST, *(current line)
2. Example ranges: 20, 20/30, ALL,FIRST/20,*/LAST, "ABCD" (next occurrence)
3. Syntax for rangelist: range, range, . . .
4. Only first letter of command name is required.
5. Q option means QUIET, except in KEEPQ(quick) and Q command(display)
6. GATHER ALL renumbers the textfile.
7. To put many commands on a line, separate them with semicolons(;).
8. Formal file designator is EDTLIST.

FCOPY (File Copier)

Operation

```
:RUN FCOPY.PUB.SYS[;INFO="FROM[={filename
                                 *
                                 <empty>}]
                    TO[={(dfile,kfile)
                         filename
                             *
                         <empty>}][;optionlist]"]
```

or

```
:FCOPY[FROM[={filename
              *
              <empty>}];TO[={(dfile,kfile)
                             filename
                                 *
                             <empty>}][;optionlist]]
>FROM [={filename
          *
          <empty>
          $CTUL
          $CTUR}];TO[={(dfile,kfile)
                       filename
                           *
                       <empty>
                       $CTUL
                       $CTUR}]
```

Specifying Carriage Control [;CCTL ;NOCCTL]

Deblocking Records [;DEBLOCK=logical-record-length]

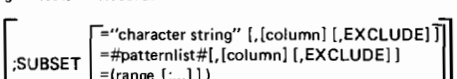
Copying Multiple Tape Files [;FILES= {number of files ALL}]

Translating Code

```
[;EBCDICIN
 ;EBCDICOUT
 ;BCDICIN
 ;BCDICOUT
 ;EBCDIKIN
 ;EBCDIKOUT
                [={field
                   (field[,field[,...]])}  [,EXCLUDE]]]
```

Omitting User Labels [;NOUSERLABELS]

Selecting Subsets of Records



Note: Record numbers start with 0. Column numbers start with 1.

FCOPY

Shifting [;UPSHIFT]

Skipping EOF

$$\left[;\text{SKIPEOF}=\left[\begin{array}{l}\{+/-\}\ \text{from-eofs}\\ \text{from-file-number}\end{array}\right]\left[,\left\{\begin{array}{l}\{+/-\}\ \text{to-eofs}\\ \text{to-file-number}\end{array}\right\}\right]\right]$$

Ignoring errors [;IGNERR [=number-of-errors]]

Creating new file [;NEW]

Verifying copy [;VERIFY[=number-of-errors]]

Comparing from file with to file [;COMPARE[=number-of-errors]]

Displaying numerical codes

$$\left[;\left\{\begin{array}{l}\text{OCTAL}\\ \text{HEX}\end{array}\right\}\left[;\left\{\begin{array}{l}\text{CHAR}\\ \text{CLEAR}\\ \text{KANA}\end{array}\right\}\right]\begin{array}{l}[;\text{NORECNUM}]\\ [;\text{TITLE="title"}]\end{array}\right]$$

Displaying Characters

$$\left[;\left\{\begin{array}{l}\text{CHAR}\\ \text{CLEAR}\\ \text{KANA}\end{array}\right\}\left[;\left\{\begin{array}{l}\text{HEX}\\ \\ \text{OCTAL}\end{array}\right\}\right]\begin{array}{l}[;\text{NORECNUM}]\\ [;\text{TITLE="title"}]\end{array}\right]$$

Determine Sequence for copying KSAM file [;KEY=character location]

Copy Data from KSAM file [;NOKSAM]

Examples

3 DISC FILES TO TAPE

```
:FILE T;DEV=TAPE
:RUN FCOPY.PUB.SYS
>FROM = file1;TO=*T
>FROM = file2;TO=*T;SKIPEOF=,2
>FROM = file3;TO=*T;SKIPEOF=,3
>EXIT
  (this requires 3 = REPLYs)
```

3 FILES FROM TAPE TO DISC

To copy back from tape –

```
:RUN FCOPY.PUB.SYS
>FROM=*T;TO=fileA;NEW
>FROM=*;TO=fileB;SKIPEOF=+1;NEW
>FROM=*;TO=fileC;SKIPEOF=+1;NEW
>EXIT
  (this assumes all files are same recsize)
```

SORT

Operation

:RUN SORT.PUB.SYS

Commands

>ALTSEQ modspec1[,modspec2] . . .[,modspecN]

modspec= [EACH / MERGE] {string spec / numbyte spec / range string spec} {= / WITH} {string spec / numbyte spec / range string spec}

>DATA [IS] {ASCII / EBCDIC} [,] **SEQUENCE** [IS] {ASCII / EBCDIC}

>END

>EXIT

>INPUT {* / $STDIN[X] / fname / (fname1,fname2, . . .,fnameN)} [,#records] [,rec size]

Note: Formal file designator is INPUT

>KEY keyspec1[;keyspec2] . . .[;keyspecN]

Notes:
keyspec=position,length[,type] [,DESC]
type may be BYTE,INT,DOUBLE,REAL,LONG,PACKED, PACKED*,DISPLAY-TRAILING-SIGN, DISPLAY-LEADING-SIGN, DISPLAY-TRAILING-SIGN-SEPARATE, or DISPLAY-LEADING-SIGN-SEPARATE

>OUTPUT {filename / * / $STDLIST} [,NUM] [,KEY]

Note: Formal file designator is OUTPUT

>RESET

>SHOW { SEQUENCE[,OFFLINE] | TABLE[,OFFLINE] | NOSEQUENCE | NOTABLE }

>VERIFY

>:[mpe command]

LIST File

Formal file designator is LIST

TEXT File

Formal file designator is TEXT

DISPLAY File

Formal file designator is DISPLOUT

INPUT File

Formal file designator is INPUT

OUTPUT File

Formal file designator is OUTPUT

Intrinsics

```
                    IA        IA          IV        IV
SORTINIT      (inputfiles, outputfiles,outputoption,reclen,
                  DV      IV      IA   IA       LP
              numrecs,numkeys,keys,altseq,keycompare,
                  P        IA        L          I
              errorproc,statistics,failure,errorparm,
                              I    I      I      O-V
              spaceallocation,parm1,parm2);

                  LA     IV
SORTINPUT     (record,length);

                  LA     IV
SORTOUTPUT    (record,length);

SORTEND;

                  IA
SORTSTAT      (statistics);

SORTTITLE;

                        IV       BA      I
SORTERRORMESS     (errorcode,message,length);

                   IV        IV     IV          IV      DV
SORTINITIALF  (inputfile,outputfile,outputoption,reclen,numrecs,
                  IV     IA      P          LP        IA
              numkeys,keys,errorsproc,keycompare,statistics,
                  L      O-V
              failure);

                   IV       IV          IV       IV     DV
SORTINITIAL   (inputfile,outputfile, outputoption,reclen,numrecs,
                  IV     LA      P          LP        LA
              numkeys,keys,errorproc,keycompare,statistics,
                  L      O-V
              failure);
```

Operation

:RUN MERGE.PUB.SYS

Commands

>ALTSEQ modspec1[,modspec2] . . .[,modspecN]

modspec= [EACH / MERGE] {string spec / numbyte spec / range string spec} {= / WITH} {string spec / numbyte spec / range string spec}

>DATA [IS] {ASCII / EBCDIC} [,]SEQUENCE[IS]{ASCII / EBCDIC}

>END

>EXIT

>INPUT {$STDIN / filename1,filename2} [,filename3]. . . [,filenameN]

>KEY keyspec1[;keyspec2] . . .[;keyspecN]

Notes:
keyspec=position,length[,type] [,DESC]
type may be BYTE,INT,DOUBLE,REAL,LONG,PACKED, PACKED*,DISPLAY-TRAILING-SIGN, DISPLAY-LEADING SIGN, DISPLAY-TRAILING-SIGN-SEPARATE, or DISPLAY-LEADING-SIGN-SEPARATE

>OUTPUT {filename / $STDLIST} [,num records] [,KEY]

Note: Formal file designator is OUTPUT

>RESET

>SHOW {SEQUENCE[,OFFLINE] / TABLE[,OFFLINE] / NOSEQUENCE / NOTABLE}

>VERIFY

>:[mpe command]

MERGE

LIST FILE

Formal file designator is LIST

TEXT FILE

Formal file designator is TEXT

DISPLAY FILE

Formal file designator is DISPLOUT

INPUT FILE

Formal file designator is INPUT

OUTPUT FILE

Formal file designator is OUTPUT

Intrinsics

```
                    IA          P              IA              P
MERGEINIT     (inputfiles,preprocessor,outputfiles,postprocessor,
                  LV         IV      IA    IA        LP
              keysonly,numkeys,keys,altseq,keycompare,
                  P          IA        L          I
              errorproc,statistics,failure,errorparm,
                          I         I        I      O-V
              spaceallocation,parm1,parm2);

                  LA      I
MERGEOUTPUT (record,length);


MERGEEND;

                    IA
MERGESTAT     (statistics);


MERGETITLE;
```

```
                                IV        BA        I
MERGEERRORMESS(errorcode,message,length);

                IV              IA          IV          IV          IV
MERGE (numinputfiles,inputfiles,outputfile,keysonly,numkeys,
          IA          P                P            P            LP
        keys,preprocessor,postprocessor,errorproc,keycompare,
            IA        L        O-V
        statistics,failure);
```

IMAGE, QUERY

SCHEMA Processor

Operation

:RUN DBSCHEMA.PUB.SYS[;PARM=n]

where

n = 1

if an actual file designator has been equated to DBSTEXT

n = 2

if an actual file designator has been equated to DBSLIST

n = 3

if actual file designators have been equated to both DBSTEXT and DBSLIST.

File Designators:

File	Use	Formal File Designator	Default Actual File Designator
textfile	Schema and Schema Processor commands	DBSTEXT	$STDINX
listfile	output listing	DBSLIST	$STDLIST

Commands

$PAGE [["character-string"], . . .]

$CONTROL [LIST / NOLIST] [,ERRORS=nnn] [,LINES=nnnnn]

[,ROOT / ,NOROOT] [,BLOCKMAX=nnnn] [,TABLE / ,NOTABLE]

$TITLE [["character-string"], . . .]

IMAGE

SCHEMA Structure

BEGIN DATA BASE data base name;
PASSWORDS: password part
ITEMS: item part
SETS: set part
END.

The form of the password part is:

user class number [password] ;
.
.
.
user class number [password] ;

The form of the item part is:

item name, [sub-item count] type designator
[sub-item length] [(read class list/write class list)] ;

The form of the set part for Master Data Sets is:

{NAME: / N:} set name, {MANUAL / M / AUTOMATIC / A}
[(read class list/write class list)] ;
{ENTRY: / E:} item name [(path count)],
.
.
item name [(path count)] ;
{CAPACITY: / C:} maximum entry count;

The form of the set part for Detail Data Sets is:

{NAME: / N:} set name, {DETAIL / D} [(read class list/
write class list)] ;
{ENTRY: / E:} item name [([!] master set name
[(sort item name)])],
.
.
item name [([!] master set name
[(sort item name)])] ;
{CAPACITY: / C:} maximum entry count;

DBLOAD

Operation

```
[:FILE DBLOAD[=filename] [;DEV=device] ]
:RUN DBLOAD.PUB.SYS
WHICH DATA BASE? data base name[/maintenance word]
DATA SET 1: x ENTRIES
        •
        •
        •
END OF VOLUME m,y READ ERRORS RECOVERED
DATA BASE LOADED
  END OF PROGRAM
```

DBRECOV

Operation

```
:RUN DBRECOV.PUB.SYS
```

Commands

```
>CONTROL     param [,param. . .]
```

param may be:

ABORTS, EOF=nnnn, ERRORS=nnnn, MODEX, MODE4, NOABORTS, NOSTAMP, NOSTATS, NOSTORE, STAMP, STATS, STOPTIME=mm/dd/yy hh:mm, or STORE

```
>EXIT

>FILE     fileref,userref[,rmode,fmode]

>PRINT  { DBTABLE   }
        { FILETABLE }

>RECOVER    data base name[/maintenance  word]
            [.group[.account] ]

>RUN
```

IMAGE

DBRESTOR

Operation

```
[:FILE DBRESTOR[=filename] [;DEV=device] [;NOBUF]]
:RUN DBRESTOR.PUB.SYS
WHICH DATA BASE? data base name[/maintenance word]
DATA BASE RESTORED
  END OF PROGRAM
```

DBSTORE

Operation

```
[:FILE DBSTORE[=filename] [;DEV=device] [;NOBUF]]
:RUN DBSTORE.PUB.SYS
WHICH DATA BASE? data base name [/maintenance word]
DATA BASE STORED
  END OF PROGRAM
```

DBUNLOAD

Operation

```
[:FILE DBUNLOAD[=filename] [;DEV=device]]
:RUN DBUNLOAD.PUB.SYS [,CHAINED]
                      [,SERIAL ]
WHICH DATA BASE? data base name[/maintenance word]
DATA SET 1:x ENTRIES
          •
          •
          •
END OF VOLUME m, y, WRITE ERRORS RECOVERED
DATA BASE UNLOADED
  END OF PROGRAM
```

DBUTIL

Operation

:RUN DBUTIL.PUB.SYS

Commands

ACTIVATE data-base-access file name

CREATE data base name[/maintenance word]

DEACTIVATE data-base-access file name

DISABLE data base name[/maint word] FOR { LOGGING | RECOVERY | ACCESS | DUMPING }

ENABLE data base name[/maint word] FOR { LOGGING | RECOVERY | ACCESS | DUMPING }

ERASE data base name[/maintenance word]

EXIT

HELP [commandname]

PURGE data base name[/maintenance word]

SET data base name [/maint word]

{ MAINT = maintenance word
BUFFSPECS = num buffers (from-users/to-users)
[,num buffers (from-users/to-users)] . . .
LOGID = log identifier
PASSWORD classnum=[password] }

IMAGE

SHOW data base name[/maint word]

{ MAINT
ALL
BUFFSPECS
LOCKS
USERS
LOGID
FLAGS
PASSWORDS } [OFFLINE]

VERIFY data-base-access file name

Intrinsics

Calling an IMAGE procedure:

COBOL	CALL "name" USING parameter,parameter, . . ., parameter
FORTRAN	CALL name (parameter,parameter, . . .,parameter)
SPL	name (parameter,parameter , . . .,parameter)
BASIC	linenumber CALL name (parameter,parameter , . . .,parameter)

Note: ALL parameters are required.

```
                A    A     I     A      I
DBBEGIN        (base,text,mode,status,textlen);
                      I or
                A    A     I     A
DBCLOSE        (base,dset,mode,status);

                A      A       I    A
DBCONTROL      (base,qualifier,mode,status);
                      I or
                A    A     I     A
DBDELETE       (base,dset,mode,status);

                A    A     I     A      I
DBEND          (base,text,mode,status,textlen);

                A     A      I
DBERROR        (status,buffer,length);

                A
DBEXPLAIN      (status);
```

Note: The base, qualifiers dset, and password parameters, if required for the procedure which put the results in the status area, must be unchanged when the call is made to DBEXPLAIN.

```
                      I or                I or
                A    A     I     A     A      A
DBFIND         (base,dset,mode,status,item,argument);
```

```
                    I or
                A   A    I    A    A    A     A or DI
DBGET          (base,dset,mode,status,list,buffer,argument);
```

Reading methods:

mode 1	Re-read
mode 2	Serial read
mode 3	Backward serial read
mode 4	Directed read
mode 5	Chained read
mode 6	Backward chained read
mode 7	Calculated read
mode 8	Primary calculated read

Status array contents (if successful):

word 1	0
word 2	Integer word length of the logical entry read into the buffer array.
words 3-4	Doubleword record number of the data entry read.
words 5-6	Doubleword zero, unless the entry read is a primary entry in which case it is the number of entries in the synonym chain.
words 7-8	Doubleword record number of the preceding entry in the chain of the current path.
words 9-10	Doubleword record number of the next entry in the chain of the current path.

```
                    I or
                A    A      I    A     A
DBINFO         (base,qualifier,mode,status,buffer);

                A    A or I    I    A
DBLOCK         (base,qualifier,mode,status);
```

Locking modes:

mode 1	Data base, unconditional
mode 2	Data base, conditional
mode 3	Data set, unconditional
mode 4	Data set, conditional
mode 5	Data entries, unconditional
mode 6	Data entries, conditional

Status array contents (if successful):

word 1	0
word 2	The number of lock descriptors that were successfully applied in the DBLOCK request. For successful locks in modes 1 through 4 this will be 1.
word 3	If condition word = 20, this word contains 0 if data base locked, 1 if data set or entries locked.
word 4	Reserved: Contents undefined.
words 5-10	Information about the procedure call and its results. (See Appendix A of the IMAGE Data Base Management Reference Manual, part no. 32215-90003.)

```
                A   A     I    A      I
DBMEMO        (base,text,mode,status,textlen);

                A     A        I     A
DBOPEN        (base,password,mode,status);
```

Access Mode	Associated Capabilities	Concurrent Modes Allowed
1	modify with enforced locking. Allow concurrent modify	1,5
2	update, allow concurrent update	2,6
3	modify exclusive	none
4	modify, allow concurrent read	6
5	read, allow concurrent modify	1,5
6	read, allow concurrent modify	6 and either 2, one 4, or 8.
7	read, exclusive	none
8	read, allow concurrent read	6,8

```
                    I or
                A   A     I    A    A    A
DBPUT         (base,dset,mode,status,list,buffer);

                    I or
                A   A     I    A
DBUNLOCK      (base,dset,mode,status);

                    I or
                A   A     I    A    A    A
DBUPDATE      (base,dset,mode,status,list,buffer);
```

Intrinsics Exceptional Conditions

Condition Code	Condition	Returned by:
10	Beginning of file	DBGET
11	End of file	DBGET
12	Directed beginning of file	DBGET
13	Directed end of file	DBGET
14	Beginning of chain	DBGET
15	End of chain	DBGET
16	Data set full	DBPUT
17	No master entry	DBFIND
17	No entry	DBGET, DBUPDATE, DBDELETE
18	Broken chain	DBGET
20	Data base locked or contains locks	DBLOCK, modes 2,4,6
22	Data set locked by another process.	DBLOCK, modes 4,6
23	Entries locked within set.	DBLOCK, mode 4
24	Item conflicts with current locks.	DBLOCK, mode 6
25	Entries already locked.	DBLOCK, mode 6
41	Critical item	DBUPDATE
42	Read only item	DBUPDATE
43	Duplicate search item value	DBPUT
44	Chain head	DBDELETE
50	Buffer too small	DBGET, DBINFO
51	Insufficient stack for BIMAGE temporary buffer	XDBGET, XDBPUT, XDBUPDATE, XDBINFO
52	Invalid number of parameters	XDBGET, XDBPUT, XDBUPDATE, XDBINFO

Condition Code	Condition	Returned by:
53 54	Invalid parameter Status array too small	XDBGET, XDBPUT, XDBUPDATE, XDBINFO
60	Data base access disabled	DBOPEN
61	This data base opened more than 63 times by same process	DBOPEN
62	DBCB full	DBGET,DBPUT, DBUPDATE DBLOCK DBBEGIN DBEND DBMEMO
63	Bad DBCB	all intrinsics
64	PCBX data segment area full	DBOPEN
66	The current DBCB for the data base does not appear correct (IMAGE internal error)	
71	Logging is disabled	DBMEMO
1xx	Missing chain head	DBPUT
2xx	Full chain	
3xx	Full master	

QUERY

Operation

:RUN QUERY.PUB.SYS

Commands:

>ADD[,] data set name

>ALTER procedure name

QUERY prompts for insert, replace, delete, or end statements by printing >>. Each statement operates on a range of lines, where **m** is the first line number and **n** is the last line number. n must be greater than or equal to m, and m must be greater than or equal to 1. Neither m nor n may exceed the total number of lines in the procedure.

Insert statement >> /I,m
>> insertion

Replace statement >> /R,m[,n]
>> replacement

Delete statement >> /D, m[,n]

End statement >> /E

Note: Terminating the ALTER command with a Y^C causes cancellation of the entire command; the procedure remains in its original state.

> ASSIGN option = { ON / OFF }

> CREATE SPACE

> CREATE procedure name, { filename / command }

> DATA-BASE = data base name

> DATA-SETS =[data set list]

> DEFINE

> **DEL**ETE

> **DES**TROY procedure name

> **DI**SPLAY procedure name { ,m[,n] / ,filename }

> **DI**SPLAY LIST

> **E**XIT

> **F**IND relation [{ AND / OR } relation . . .] [END]

where relation takes the form:
[data set name.] data item name relop "value"[,"value". . .]

Relational Operators

Operator	Meaning
= IS IE EQ	is equal to
# ISNOT INE NE < >	is not equal to
ILT LT <	is less than
INLT GE >=	is not less than (is greater than or equal to)
IGT GT >	is greater than
INGT LE <=	is not greater than (is less than or equal to)
IB $value_1$,$value_2$	is between (and including) $value_1$ and $value_2$

multiple values may be used

> FIND ALL [data set name.] data item name

> FIND CHAIN item identifier { IS / IE / EQ / = } "value"

[{ AND / OR } item identifier { IS / IE / EQ / = } "value" . . .] [END]

> FIND procedure name [,character]

> FORM [data set name / data item name / SETS / ITEMS / PATHS]

> HELP [command name]
[FUNCTION] [FORMAT] [PARAMETERS]

> LIST { data set name / data item list }

[FOR relation [{ AND / OR } relation. . .]] [END]

See FIND command for definition of relation and relational operators.

> MODE = mode number

Access Mode	Associated Capabilities	Concurrent Modes Allowed
1	modify with enforced locking. Allow concurrent modify	1, 5
2	update, allow concurrent update	2, 6
3	modify exclusive	none
4	modify, allow concurrent read	6
5	read, allow concurrent modify	1, 5
6	read, allow concurrent modify	6 and either 2, one 4, or 8
7	read, exclusive	none
8	read, allow concurrent read	6, 8

> **OUTPUT** = { TERM / LP }

> **PA**SSWORD = password

> **PR**OC-FILE = filename [,n]

> **REN**AME old procedure name, new procedure name

> **REPL**ACE, data item name="value";
[data item name="value"; . . .] END

QUERY

>REPORT report statements END

QUERY prompts for statements by printing >>.

Statements:

Detail

D[detail number], print element, print position
[,SPACE A[number]] [,SPACE B[number]]

[,SKIP {A / B}] [,E {number / Z}]

Edit

E number, "edit mask"

Group

G level, print element, print position
[,SPACE A[number]] [,SPACE B[number]]

[,SKIP {A / B}] [,E {number / Z}]

Header

H header number, print element, print position
[,SPACE A [number]] [,SPACE B[number]]

Output Control

[LINES = integer
  NOPAGE
  [OUT =] LP
  PAUSE]

Register

R number, [LOAD / ADD / SUBTRACT / MULTIPLY / DIVIDE] ,data element

Sort

S[level],data item name [, {ASC / DES}]

Total

T level, print element, print position
[,SPACE A[number]] [,SPACE B[number]]

[,SKIP {A / B}]

[,E {number / Z}] [, { ADD / AVERAGE / COUNT }]

OR

T level, Rn

Statement Parameters

PARAMETER	FUNCTION
print position	determines the rightmost print position (column number) for the print element. For character data, this is the rightmost character; for numeric data, it is the position of the least significant digit.
SPACE A[number]	Space number lines after printing the report line. If number is omitted, one line is spaced.
SPACE B[number]	Space number lines before printing the report line. If number is omitted, one line is spaced.
number	is the number of lines to be spaced (from 1 to 5)
SKIP {A / B}	Skip to the top of the next report page after printing the report line (SKIP A) or before printing the report line (SKIP B).
E {number / Z}	indicates that either an edit mask defined in the identically numbered edit statement (Enumber) is to be used to punctuate a value or, if you use the letter Z, that leading zeros are to be suppressed. In the latter case, no edit statement is required.

> **R**EPORT [output control statements*] ALL[,character]

> **R**EPORT [output control statements*] procedure name [,character]

> **SH**OW option

> **U**PDATE **ADD**, data set name

*See REPORT command (above) for output control statements.

> UPDATE **DELETE**

> UPDATE **REPLACE**, data item name = "value";[data item name = "value";. . .] END

> UPDATE procedure name [,character]

> VERSION

> XEQ filename [,NODATA]

KSAM, V/3000

KSAMUTIL

Operation

:RUN KSAMUTIL.PUB.SYS

Commands

> **B**UILD filereference 1
[;REC=[recsize] [,[blockfactor] [, [F|V] [,BINARY|,ASCII]]]]
[;TEMP]
[;DEV=device]
[;CODE=filecode]
[;DISC=[numrec] [,[numextents] [,initialloc]]]

;KEYFILE=filereference 2

;KEY=keytype,keylocation,keysize[,[keyblocking]
[,**DUP**LICATE | ,**RDUP**LICATE]]

[;KEY=keytype,keylocation,keysize[,[keyblocking]
[,**DUP**LICATE | ,**RDUP**LICATE]]]

[;KEYENTRIES=numentries]
[;LABELS=numlabels]
[;KEYDEV=device]
[;FIRSTREC=recnum]

Note: keytype may be: **B**YTE
INTEGER
DOUBLE
REAL
LONG
NUMERIC
PACKED
***P**ACKED

filereference is an actual file designator

KSAM

>ERASE filereference

>EXIT

>HELP

>KEYDUMP [filereference] [;SEQ=keysequence]

[;FILE=formaldesignator] [;SORT]
[;OFFLINE]

>KEYINFO [filereference] [;OFFLINE] [;RECOVER]

>KEYSEQ [filereference] [;SEQ=keysequence] [;OFFLINE]
[;NOLIST]

>PURGE filereference[,TEMP]

>RENAME oldfilereference,newfilereference[,TEMP]

>SAVE [filereference]

>VERIFY [filereference] [;OFFLINE] [;NOCHECK]

SPL Intrinsics

Format:

intrinsicname (parameterlist);

or

return:=intrinsicname (parameterlist);

Example:

FCLOSE(FILENUM,DISP,SECCODE);

or

LNGTH:=FREAD(FILENUM,TARGET,COUNT);

FCHECK
IV I I D I O-V
(**filenum**, errorcode, tlog, blknum, numrecs);

Condition Codes: CCE, CCL

FCLOSE
IV IV IV
(**filenum, disposition, seccode**);

Condition Codes: CCE, CCL

Note: disposition (12:1) should = 0 for KSAM files

FCONTROL
IV IV **L**
(**filenum, controlcode, param**);

Condition Codes: CCE, CCL

Note: controlcode value for KSAM files may be:

2 - Complete all I/O

5 - Rewind File

6 -Write MPE EOF

7 - Clear buffers

The param parameter included for compatibility only

PL Intrinsics (continued)

```
                    I      LA        I
ERRMSG      (errorcode, msgbuf, msglnth);

                IV       BA          IV          IV    IV
FINDBYKEY   (filenum, keyvalue, keylocation, keylength, relop);
```

Condition Codes: CCE, CCG, CCL

```
                IV      DV          IV
FINDN       (filenum, number, keylocation);
```

Condition Codes: CCE, CCG, CCL

```
                IV       BA        L         L       I
GETINFO     (filenum, filename, foptions, aoptions, recsize,

                I       L        L         I      D    D
            devtype, ldnum, hdaddr, filecode, recptr, eof,

                D         D          D        I       L
            flimit, logcount, physcount, blksize, extsize,

                        I          I       BA        D  O-V
            numextents, userlabels, creatorid, labaddr);
```

Condition Codes: CCE, CCL

Note: filename parameter must be 27 bytes long.

```
                IV          A            A
GETKEYINFO  (filenum, ksamparam, ksamcontrol);
```

Condition Codes: CCE, CCL

```
                IV       LV
LOCK        (filenum, lockcond);
```

Condition Codes: CCE, CCG, CCL

```
FOPEN          I                                  BA      LV
               filenum := FOPEN (formaldesignator, foptions,

                   LV      IV     BA        BA         IV
               aoptions, recsize, device, ksamparam, userlabels,

                    IV          IV      DV          IV
               blockfactor, numbuffers, filesize, numextents,

                   IV      IV     O-V
               initialloc, filecode);
```

foptions:

(0:4)	(4:1)	(5:1)	(6:1)	(7:1)	(8:2)	(10:3)	(13:1)	(14:2)
Reserved	**KSAM File**	**Disallow :FILE**	**Reserved**	**Carriage Control**	**Record Format**	**Default Designator**	**ASCII/ Binary**	**Domain**
	0≡not a new KSAM file 1≡new KSAM file or opened as MPE file	0≡Allow :FILE 1≡No :FILE		0≡NOCCTL	00≡Fixed 01≡Variable	000≡filename	0≡Binary 1≡ASCII	00≡New file 01≡Old Permanent File 10≡Old Temporary File 11≡Old Perm. or Temp. File

aoptions:

(0:3)	(3:1)	(4:1)	(5:3)	(8:2)	(10:1)	(11:1)	(12:4)
Reserved	**KSAM Access**	**No-Wait I/O**	**Reserved**	**Exclusive Access**	**Dynamic Locking**	**Reserved**	**Access Type**
	0≡KSAM access 1≡non-KSAM access	0≡No-Wait		00≡Default 01≡Exclusive 10≡Exclusive Access Read 11≡Share	0≡No FLOCK Allowed 1≡ FLOCK Allowed		0 000≡Read Only 0 001≡Write only 0 010≡Write (save) only 0 011≡Append only 0 100≡Read write 0 101≡Update 0 110≡Execute

```
                    IV      DV
FPOINT        (filenum, recnum);
```

Condition Codes: CCE, CCG, CCL

```
FREAD
        I                  IV    LA      IV
      lgth: = FREAD (filenum, target, tcount);
```

Condition Codes CCE, CCG, CCL

```
FREADBYKEY
        I                          IV    LA      IV        BA
      lgth: = FREADBYKEY (filenum, target, tcount, keyvalue,

                    IV
            keylocation);
```

Condition Codes: CCE, CCG, CCL

```
FREADC
        I                  IV    LA      IV
      lgth: = FREADC (filenum, target, tcount);
```

Condition Codes: CCE, CCG, CCL

```
                    IV    LA      IV      DV
FREADDIR      (filenum, target, tcount, recnum);
```

Condition Codes: CCE, CCG, CCL

```
                    IV    LA      IV      IV   O-V
FREADLABEL    (filenum, target, tcount, labelid);
```

Condition Codes: CCE CCG, CCL

```
                    IV
FREMOVE       (filenum);
```

Condition Codes: CCE, CCG, CCL

```
                    IV          LV
FSETMODE      (filenum, modeflags);
```

Condition Codes: CCE, CCL

Note: Only bit 14 of modeflags is used:

(14:1) = 1 - Verify output

= 0 - Do not verify output

```
                        IV          IV
FSPACE          (filenum, displacement);

                Condition Codes: CCE, CCG, CCL

                        IV
FUNLOCK         (filenum);

                Condition Codes: CCE, CCG, CCL

                        IV    LA    IV
FUPDATE         (filenum, target, tcount);

                Condition Codes: CCE, CCG, CCL

                        IV    LA    IV    LV
FWRITE          (filenum, target, tcount, control);

                Condition Codes: CCE, CCG, CCL

                        IV    LA    IV    IV  O-V
FWRITELABEL     (filenum, target, tcount, labelid);

                Condition Codes: CCE, CCG, CCL

HP32208
        D
   version:= HP32208
```

KSAM

COBOL Procedures

Format: CALL "procedurename" USING parameterlist

Example: CALL "CKREAD" USING FILETBL, STAT, REC, RECSIZE.

CKCLOSE **filetable status**

CKDELETE **filetable status**

CKERROR **status result**

CKLOCK **filetable status lockcond**

CKOPEN **filetable status**

CKOPENSHR **filetable status**

CKREAD **filetable status record recordsize**

CKREADBYKEY **filetable status record key keyloc recordsize**

CKREWRITE **filetable status record recordsize**

CKSTART **filetable status relop key keyloc keylength**

CKUNLOCK **filetable status**

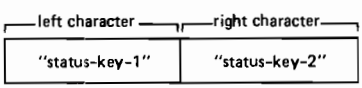
CKWRITE **filetable status record recordsize**

Note: All parameters are required, and must be separated by at least one space,

filetable parameter:

Word		
1	filenumber	
2		
3	filename (8 characters)	
4		
5		
6	input-output type	
7	access mode	
8	lock/unlock	previous operation

status parameter:



If left character of status (status-key-1) equals:	Then right character of status (status-key-2) may equal:
"0" (successful completion	"0" (no further information)
	"2" (duplicate key)
"1" (at end)	"0" (no further information)
"2" (invalid key)	"1" (sequence error)
	"2" (duplicate key)
	"3" (no record found)
	"4" (boundary violation)
"3" (request denied)	"0" (lock denied)
	"1" (unlock denied)
"9" (file system error)	"*n*" where *n* is the MPE file system error code.

KSAM

BASIC Procedures

Format: statement label CALL procedurename (parameters)

Example: 250 CALL BKCLOSE(FILENUM,STAT)

BKCLOSE **filenum, status**

BKDELETE **filenum, stetus**

BKERROR **status, message**

BKLOCK **filenum, status,** condition

BKOPEN **filenum, status, filename,** access, dynamic lock, exclusive, sequence

BKREAD **filenum, status**, parameterlist

BKREADBYKEY **filenum, status, keyvalue, keylocation, paremeterlist**

BKREWRITE **filenum, status, parameterlist**

BKSTART **filenum, status,** keyvalue, keylocation, relation

BKUNLOCK **filenum, status**

BKVERSION **status, message**

BKWRITE **filenum, status, parameterlist**

Note: parameterlist is a list of variables into which data is read (BKREAD, BKREADBYKEY) or a list of variables or constants containing data to be written (BKREWRITE, BKWRITE).

status parameter:

FIRST CHARACTER	REMAINING CHARACTERS
"0" successful completion	"0" no further information
	"2" duplicate key value
"1" at end or beginning of file	"0" no further information
"2" invalid key	"1" sequence error
	"2" duplicate key error
	"3" no record found
	"4" boundary violation
"7" request denied	"1" file already locked
"8" invalid call	"1" invalid number of parameters
	"2" invalid parameter
	"3" insufficient space for data in *parameterlist*
"9" file system error	"0" through "255" corresponding to file system error codes

FORTRAN Procedures

Format: CALL procedurename (parameters)

Example: CALL FCLOSE(FILENO,DISP,CODE)
or
CKCLOSE(FILETAB,STAT)

Note: FORTRAN may call either the SPL intrinsics or the COBOL procedures.

FORMSPEC

:RUN FORMSPEC.PUB.SYS

Function Keys

f1	f2	f3	f4
PREV FORM	NEXT FORM	FIELD TOGGLE	REFRESH
PREV	NEXT	MAIN/RESUME	EXIT
f5	f6	f7	f8

Identifiers

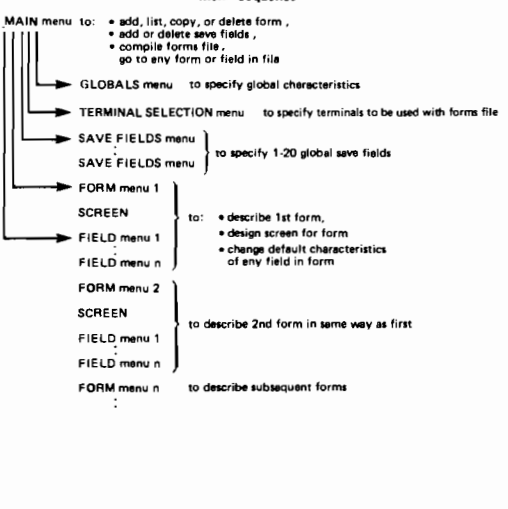
Forms File: data file name: filename[/lockword] [.groupname[.accountname]]
key file name: filename

Form Name: 1-15 character uppercase name unique to forms file; 1st character A-Z, other characters A-Z, 0-9, or underline (_)

Field Tag: Upper or lowercase field identifier unique to form; entered within field delimiters; otherwise, like form name

Field Name: Uppercase field tag or alternate field name unique to form; specified exactly like form name

Menu Sequence



Global Definitions

Head form: Any form name; *default* = 1st form in file

Enhancement I = Inverse Video
U = Underline
B = Blinking
H = Half Bright
S = Security
NONE = no enhancements

Default Enhancements: **Field** = IH (Inverse Video, Half Bright)
Error = IU (Inverse Video, Underline)
Window = IH (Inverse Video, Half Bright)

Window Line: Any screen line number from 1-24 or zero for no window *default* = line 24 (bottom line)

Define Function Key Labels: Y = obtain the Global Function Key Labels Menu to change the function key labels. To retain the default Global labels, leave this field blank.

Form Definition

Form Name: Any form name unique to forms file

Next Form Name: Any form name in file, or $END, $HEAD, $RETURN, $REFRESH

Repeat Option: N = do not repeat current form
A = repeat current form, appending to previous form
R = repeat current form, overlaying previous form

Next Form Option: C = clear current form before displaying next form
A = append next form to current form
F = freeze current form, then append next form

Reproduced From: If you are generating a son form, enter the name of the parent form here. (Optional.)

Comments: Enter any comment up to 50 characters long.

Local Form Function Key Labels: Y = obtain the Form Functions Key Labels menu and change function key labels. To retain local default labels, leave this field blank.

Form Function Key Labels

Function Key: For each Function Key, enter the first line of the label in the first field and the second line of the label in the second field. Each field may contain up to eight characters.

Global Function Key Labels

Function Key: For each Function Key, enter the first line of the label in the first field and the second line of the label in the second field. Each field may contain up to eight characters.

Screen Design

Field Delimiters: Printing: start field = [(open bracket)
stop field =] (close bracket)
Non-printing: start field = CNTL/f2 or ESC [
stop field = CNTL/f3 or ESC]

Field Tag: Upper or lowercase name within field delimiters

Field Length: Number of characters within field delimiters

Unprotected Field Enhancement: Determined by default or global specifications; may be changed for individual fields on Field menu

Protected Area Enhancement: Use terminal enhancement codes: CNTL/f1 followed by uppercase letter for enhancement:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
							X	X	X	X	X	X	X	X	Half Bright
			X	X	X	X					X	X	X	X	Underline
	X	X			X	X			X	X			X	X	Inverse Video
X		X		X		X		X		X		X		X	Blinking

or @ to clear enhancement

Field Definition

Field Number: Number assigned permanently to each field in order of initial field definition, starting with 1

Field Name: By default, upshifted field tag; may be changed to any upper-case field name unique to form

Field Type:
O = Optional field; skip edit checks if blank (*default*)
R = Required field; error if blank
D = Display-only field; protected from operator entry
P = Process field; perform edit checks even if blank

Data Type:
CHAR = Any ASCII characters allowed (*default*)
NUM = Floating-point signed number, commas allowed
NUM*n* = Fixed-point signed number, maximum *n* decimal places, commas allowed
IMP*n* = Implied decimal point signed number, *n* decimal positions, commas allowed
DIG = Digits only allowed
MDY = Date in order: month day year
DMY = Date in order: day month year
YMD = Date in order: year month day
month = 1 or 2 digits in range 1-12, 3-letter abbreviation, or full name of month, correctly spelled.
day = 1 or 2 digits in range 1-31
year = 2 or 4 digits of legitimate year
separator = / or - or . or blank (if blank, month and day must be 2 digits)

Field Enhancement: I, B, U, H, S (in any combination) or NONE (*default* = IH)

Initial Value: Any value that matches field's data type, length

Save Field Definition

Save Field Name: Field name unique to forms file

Save Field Data Type: Any legal type: CHAR, NUM, NUM*n*, IMP*n*, DIG, MDY, DMY, or YMD

Save Field Length: Maximum number of characters in field

Initial Value: Any value of field's data type and within field length

Terminal Selection

Enter X in the fields by the terminals which will be used with the forms file. Default: the forms file will run on the 264X family and the 262X family, but local edits and security display enhancements on the 262X family will not be supported.

3075/76 Globals

Split Message Pause: Integer; number of seconds to retain a line of text on the screen.

Wait for user to press enter: Enter an "X" in this field to retain a line of text on the screen until user presses "enter".

Error light: Character; indicates which key's light will be turned on when errors are detected. (Default: E.)

HOLES/MARKS: Tells whether multiple function reader will be reading cards with holes or marks. (Default: holes.)

Corner Cut Required: Enter YES or NO. (Default: YES.)

Clock On/After/NONE Whether clocking marks occur at the same time as data, after the data, or don't occur at all. (Default: NONE.)

Barcode Reader Format: Enter UPC, EAN, I39, I25, or MAT. (Default: UPC.)

Reserved Words

ALL	DISPLAY	I25	LOCALEDITS	NONE	TRAILING
APPEND	EAN	I39	LT	NOREPEAT	TYPEV
BARCODE	ELSE	IF	MAGSTRIPE	OF	UPC
CAD	EQ	IN	MARKS	PRINTER	UPSHIFT
CDIGIT	FAIL	INIT	MAT	REPEAT	$EMPTY
CENTER	FIELD	JUSTIFY	MATCH	RIGHT	$END
CFORM	FILL	KEYBOARD	MFR	SET	$HEAD
CHANGE	FINISH	LARGECHAR	MINLEN	STOCHAR	$LENGTH
CLEAR	FREEZE	LE	NE	STOP	$REFRESH
COD	GE	LEADING	NFORM	STRIP	$RETURN
CONFIG	GT	LEFT	NIN	THEN	$STATE
CUT	HOLES	LIGHT	NOCUT	TO	$TODAY
DEVICE					

Operands

Field Name: Name of existing field in current form

Save Field Name: Name of save field in current forms file

Constant: Character, numeric, or date type value; or system defined constant

- **Character Type:** Any ASCII characters within single or double quotes (" or ')
- **Numeric Type:** Digit string with optional leading sign and optional decimal point
- **Date Type:** Any legal date format in order MDY, enclosed within exclamation points (!)
- **System Defined:**
  $EMPTY (any data type) = null value
  $LENGTH (numeric type) = current field length
  $STATE (character type) = 2-character state abbreviation
  $TODAY (date type) = today's date
  $END (form name) = terminate forms
  $HEAD (form name) = first form
  $REFRESH (form name) = clear form
  $RETURN (form name) = previous form

Arithmetic Expression: Any numeric contents, fields, save fields, or index retrieve operands in parentheses combined by the following operators:

+ (add) - (subtract) * (multiply) / (divide) % (percent of)

Index Retrieve: Result of evaluating operand in form:

index OF **element** [,**element**] . . .

where **index** = numeric field, save field, or arithmetic expression

element = constant, field, save field, or arithmetic expression

Enclosed within parentheses in edit statements or as operand in arithmetic expression

FORMSPEC STATEMENTS

Config Statements:

```
DEVICE  [PRINTER]
        [KEYBOARD]
        [MAGSTRIPE]
        [TYPEV]

[MFR [COD MARKS  [CUT  ]]]
     [CAD MARKS  [NOCUT]]
     [NONE HOLES]
     [CAD HOLES ]

[DISPLAY [LARGECHAR]]
         [STDCHAR  ]

[BARCODE [I25 ]]
         [I39 ]
         [EAN ]
         [UPC ]
         [MAT ]
```

Note: More than one option may be selected; separate options with commas.

```
LIGHT {@                    }
      {[A] [B] [C] ... [N] [O] [P]}
```

Note: More than one character may be selected; separate characters with commas.

```
LOCALEDITS [ALPHABETIC      ]
           [ALPHA_NUMERIC   ]
           [CONSTANT        ]
           [DEC_DIGITS n    ]
           [DEC_TYPE_EUR    ]
           [DEC_TYPE_US     ]
           [IMP_DEC         ]
           [IMP_DEC_FILL    ]
           [INTEGER         ]
           [INTEGER_FILL    ]
           [JUSTIFY         ]
           [MUST_FILL       ]
           [REQUIRED        ]
           [SIGN_DEC        ]
           [SIGN_DEC_FILL   ]
           [TRANSMIT_ONLY   ]
           [UNRESTRICTED    ]
           [UPSHIFT         ]
```

Note: More than one option may be selected, separate options with commas or blanks.

Edit Statements:

CDIGIT { 10 / 11 } [**"message"**]

{ **GT** / **LT** / **GE** / **LE** / **EQ** / **NE** } **operand** [**"message"**]

{ **IN** / **NIN** } { **operand**[,**operand**] ... / **lowoperand**:**highoperand**[,**lowoperand**:**highoperand**] ... } [**"message"**]

MATCH pattern [**"message"**]

Pattern Characters:

Each character in pattern is *generic* or *actual*; generic characters determine *type* of character to enter in that position; actual characters indicate exact character to enter in that position.

Generic Characters:

a	upper or lowercase letter (A-Z, a-z)
u	uppercase letter (A-Z)
l	lowercase letter (a-z)
b	blank (space)
d	digit (0-9)
?	any character

Pattern Operators:

!	Transparency (!d means enter "d", not a digit)
,	Choice (a,d allows any letter or a digit)
:	Range (1:5 means enter one of the digits 1, 2, 3, 4, 5)
{ }	Grouping - Required ({a,d} allows 1 letter or 1 digit)
[ ]	Grouping - Optional ([a,d] allows 1 letter, 1 digit, or nothing)
+	Repetition - Required (d+ means enter as many digits as desired)
*	Repetition - Optional (d* means enter as many digits as desired, or nothing)

Note: Pattern must not require leading or trailing blanks in data since such blanks are stripped before the match is executed.

MINLEN operand [**"message"**]

Form Sequence Statements:

CHANGE CFORM TO { **NOREPEAT** / **REPEAT** / **REPEAT APPEND** }

CHANGE NFORM TO { [**CLEAR** / **APPEND** / **FREEZE APPEND**] [**"formname"** / **field name** / **index retrieve operand** / **$END** / **$HEAD** / **$REFRESH** / **$RETURN**] }

Control Statements:

FAIL [**"message"**]

STOP

Conditional Statements:

IF [operand] **editstatement THEN** [**statement**]...
[**statement**]...
[**ELSE** [**statement**]...]
[[**statement**] ...]

Data Movement Statements

SET { destination **TO** source }
{ destination }
{ **TO** source }

source = field name
save field name
constant
arithmetic expression
index retrieve operand

destination = field name
save field name

Data Formatting Statements:

FILL { **TRAILING** } **"character"**
{ **LEADING** }

JUSTIFY { **LEFT** }
{ **RIGHT** }
{ **CENTER** }

SET TO thisfield

Format entered data to default format according to data type:

Character type: No default formatting

Numeric types: Right justify, replace leading zeros with blanks, strip any commas; strip any plus sign or float minus sign; and depending on type:

NUM - try to fit 9 decimal places, then strip trailing decimal zeros

NUM*n* - insert decimal point and if needed, fill all fractional positions with zeros

IMP*n* - strip any decimal point and if needed, fill all fractional positions with zeros

Date types: Format as dd/dd/dd in order MDY, DMY, or YMD depending on field's date type; left justify, filling with blanks on right.

STRIP { TRAILING / LEADING / ALL } "characters"

UPSHIFT

Conversion Between Data Types

Source Type	Destination Type					
	CHAR	NUM	NUM*n*	IMP*n*	DIG	DATE
CHAR	truncate or pad with blanks on right	illegal	illegal	illegal	illegal	illegal
Any Numeric Type	truncate or pad with blanks on right	right justify; replace leading zeros with blanks; pad trailing blanks; strip any commas				illegal
		try to fit 9 decimals places, round or truncate, insert decimal as needed	round or truncate fractions, insert decimal point as needed; fill trailing zeros	remove any decimal point; may strip leading fractional zeros	value must be positive, round to integer	
Any Date Type	truncate or pad with blanks on right	illegal	illegal	illegal	illegal	left justify; pad trailing blanks; convert to dd/dd/dd in destination order

Phase Specifications

Config							
Initialize				INIT statements	INIT statements	[INIT] statements	[INIT] statements
Field Edit		[FIELD] statements	[FIELD] statements			FIELD statements	FIELD statements
Finish	FINISH statements		FINISH statements		FINISH statements		FINISH statements

Note: The Config phase is optional.

REFSPEC

:RUN REFSPEC.PUB.SYS

Function Keys

f1	f2	f3	f4
PREV REFORMAT	NEXT REFORMAT		REFRESH
PREV	NEXT	MAIN/RESUME	EXIT
f5	f6	f7	f8

Identifiers

Reformat File: data file name: filename[/lockword] [.groupname[.accountname]]
key file name: filename

Forms File: Name of data file created through FORMSPEC

Reformat Id: 1st form name in each input forms sequence

Menu Sequence

REFORMAT FILE menu to specify name of reformat file
FORMS FILE menu to specify forms file for reformatting
MAIN menu to:
- add, list, or delete a reformet
- list entire reformat file
- compile reformat file
- go to existing reformat or output field
- go to FORMS FILE menu

GLOBALS menu to specify globals

INPUT FORMS menu 1
OUTPUT RECORD menu 1
OUTPUT FIELD menu 1
:
OUTPUT FIELD menu n

to:
- define 1st reformat (input sequence
- specify output records for 1st reformat
- further define each field in output record

INPUT FORMS menu 2
OUTPUT RECORD menu 2
OUTPUT FIELD menu 1
:
OUTPUT FIELD menu n

to define 2nd reformat in same way

INPUT FORMS menu n to define subsequent reformets
:

Global Specifications

Output Record Format: F - Fixed-length records (*default*)
V - Variable-length records
U - Undefined-length records

Record Length

Specify total number of characters in output record; *default* = 80 characters

Upshift: Y - Shift letters to uppercase
N - Do not shift letters (*default*)

Convert to EBCDIC: Y - Write output record in EBCDIC
N - Leave data in ASCII code (*default*)

Record Terminator:

Specify terminator for each output record as 1 or more constants:

- Quoted string of ASCII characters; and/or
- Numeric equivalent of ASCII code preceded by $; and/or
- System constant:
  - $LF - line feed
  - $CR - carriage return
  - $GS - group separator
  - $US - unit separator
  - $RS - record separator

Field Separator

Separate fields in output record with any of the constants listed above under **Record Terminator**.

Reformat Specifications

Input Forms Sequence:

At least one and up to 10 form names; 1st name (*reformat id*) must be unique. Only data from forms in same order as forms in input forms sequence is reformatted.

Output Record:

Name each field, portion of field, or constant to be written to output record; input fields must be in a form listed in associated input forms sequence.

Input Field	Name of field in form in input forms sequence (required)
Substring Start	1st character (offset) in input field (optional)
Input Length	Number of characters in input field counting from column 1 of field or Substring Start (optional)
Form Name	Name of form containing input field (optional)
Output Field	Unique name for output field (optional)
Starting Column	Column number in output record where output field starts (optional)
Length	Number of characters in output field (optional)
Start of Record	Character to indicate that this is 1st field, or constant in output record (optional)
Constant	Constant to be written to output record; may be constant listed under **Record Terminator** (optional)

Output Field:

Each field in output record may be formatted according to:

Data Type - Specify type to which data in input field is to be converted; *default* = CHAR
Allowed conversions are:
Any data type to CHAR
Any numeric type to any numeric type
Any date type to any date type

REFSPEC STATEMENTS

CHECKDIGIT { 10 / 11 }

FILL { ALL / LEADING / TRAILING } character

JUSTIFY { RIGHT / LEFT / CENTER }

SIGN { LEFT / RIGHT / ZONE / FLOAT / NONE } PLUS (Y or N)

STRIP { ALL / LEAOING / TRAILING } "characters"

REFORMAT

:FILE REFFILE=reformatfile	Specify name of file containing reformat specifications entered by REFSPEC.
:FILE BATCH=batchfile	Specify name of file containing data to be reformatted.
:FILE OUTFILE=outputfile	Specify name of file to which reformatted data is written.
:FILE REFLIST;DEV=LP	List error messages on line printer, not at terminal (optional)
:FILE TESTLIST;DEV=LP	List reformatted data (optional)
:RUN REFORMAT.PUB.SYS	Run program REFORMAT

Procedures

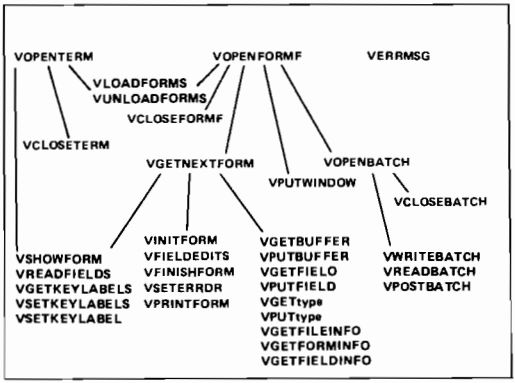
Formats:

COBOL:	CALL "procedurename" USING parameterlist
FORTRAN:	CALL procedurename (parameterlist)
SPL:	procedurename (parameterlist);
BASIC:	statement label CALL procedurename (parameterlist)

VCLOSEBATCH	comarea
VCLOSEFORMF	comarea
VCLOSETERM	comarea
VERRMSG	comaree, buffer, buflen, actualen
VFIELDEDITS	comarea
VFINISHFORM	comarea
VGETBUFFER	comarea, buffer, buflen
VGETFIELD	comarea, fieldnum, fieldbuf, buflen, actualen, nextfldnum
VGETFIELDINFO	comarea, infobuf, infobuflen
VGETFILEINFO	comarea, infobuf, infobuflen
VGETFORMINFO	comarea, infobuf, infobuflen
VGETKEYLABELS	comarea, formorglob, numoflabels, labels Note: formorglob = 0 retrieves global labels; formorglob = 1 retrieves current form labels.
VGETNEXTFORM	comarea
VGETtype	comarea, fieldnum, variable Note: type may be INT, DINT, REAL, or LONG
VINITFORM	comarea
VLOADFORMS	comarea, numofforms, formsloaded, forms Note: numofforms = –1: workspace configuration under user control. numofforms = 0: no local form storage. numofforms = 1 to 4: One to four forms can be stored locally.
VOPENBATCH	comarea, batchfile
VOPENFORMF	comarea, formfile
VOPENTERM	comarea, termfile
VPOSTBATCH	comarea
VPRINTFORM	comarea, printcntl, pagecntl

VPUTBUFFER	comarea, buffer, buflen
VPUTFIELD	comarea, fieldnum, fieldbuf, buflen, actualen, nextfldnum
VPUTtype	comarea, fieldnum, variable
VPUTWINDOW	comarea, message, msglen
VREADBATCH	comarea
VREADFIELDS	comarea
VSETERROR	comarea, fieldnum, message, msglen
VSETKEYLABEL	comarea, formorglob, keynum, label Note: formorglob = 0 sets global label; formorglob = 1 sets current form label.
VSETKEYLABELS	comarea, formorglob, numoflabels, labels Note: formorglob = 0 replaces global labels; formorglob = 1 replaces current form labels.
VSHOWFORM	comarea
VUNLOADFORM	comarea, whichform
VWRITEBATCH	comarea

Procedure Dependencies



COMAREA

Used by all V/3000 procedures, this area must be initialized to zero and COMAREALEN specified (currently 60 words) before 1st procedure call; if applicable, set LANGUAGE (except COBOL) and USRBUFLEN (BASIC).

Type	Offset SPL	Offset Other	#Words	Name	Content
1-word integer	0	1	1	CSTATUS	Status or error code; 0=successful
	1	2	1	LANGUAGE	0=COBOL; 1=BASIC; 2=FORTRAN; 3=SPL
	2	3	1	COMAREALEN	Total length of COMAREA (in words)
	3	4	1	USRBUFLEN	BASIC only; length of COMAREA extension
	4	5	1	CMODE	0=Collect; 1=Browse/Modify
	5	6	1	LASTKEY	Code of last key pressed at terminal: 0=ENTER 1=f1 2=f2 3=f3 4=f4 5=f5 6=f6 7=f7 8=f8
	6	7	1	NUMERRS	Number of edit errors found in current form
	7	8	1	WINDOWENH	ASCII code for window enhancement; initialized to FORMSPEC value, can be changed to particular code (see codes under screen design)
	8	9	1	MULTIUSAGE	Next form flag 1=Son or brother to previous form; 0=Otherwise
	9	10	1	LABEL'OPTION	0=No labels are used; 1=Labels are to be used.
Character	10	11	8	CFNAME	Name of current form (15 characters)
	18	19	8	NFNAME	Name of next form (15 characters)
1-word integer	26	27	1	REPEATAPP	Repeat/Append flag for current form; 0=Normal sequence, no repeat/append 1=Repeat current form 2=Repeat and append current form
	27	28	1	FREEZAPP	Freeze/Append flag for next form: 0=Clear current form, display next 1=Append next form to current form 2=Freeze current form, append next
	28	29	1	CFNUMLINES	Number of lines in current form
	29	30	1	DBUFLEN	Total number of characters needed in data buffer for all concatenated fields (including display-only) in current form
	30	31	1	Reserved – leave 1 word initialized to zero	

Type	Offset SPL	Offset Other	#Words	Name	Content
1-word unsigned integer	31	32	1	LOOK'AHEAD	Form preload indicator. 0- ON – Preload the forms. 1- OFF – Do not preload the forms.
	32	33	1	DELETEFLAG	Flag to delete current batch record: FALSE (all zeros) – do not delete TRUE (all 1's) – delete record
	33	34	1	SHOWCONTROL	Override VSHOWFORM optimizations; bit 15=1 write form only to terminal, 14=1 write only data from buffer, 13=1 write only window line
	34	35	1	Reserved – leave 1 word initialized to zero	
1-word integer	35	36	1	PRINTFILNUM	MPE file # of VPRINTFORM list file
	36	37	1	FILERRNUM	MPE file error # returned by procedures
	37	38	1	ERRFILENUM	MPE file # of VIEW error message file
	38	39	1	FORM'STORE'SIZE	Number of forms in form storage buffer. -1 = Workspace configuration under user control, 0 = No local form storage; 1 . 4 = One to four forms can be stored locally.
	39	40	3	Reserved – leave 3 words initialized to zero.	
2-word integer	42	43	2	NUMRECS	Number of non-deleted records in batch file
	44	45	2	RECNUM	Record number of current batch record, records are counted from zero
	46	47	2	Reserved – leave 2 words initialized to zero	
Logical	48	49	1	FILEN	MPE file number of terminal
	49	50	5	Reserved – leave 5 words initialized to zero	
Logical	54	55	1	RETRIES*	Maximum number of retries. value = 0 – use default (4 retries); value > 0 – use this value as maximum value < 0 – no retries
	55	56	1	OPTIONS*	Terminal control options: bits: 0–8 reserved for system use; 9–10 – 01 enables ENTER/FCN key timeout; 11 or 00 disables ENTER/FCN key timeout. 13–14 – 01 enables AUTOREAD; 11 or 00 disables AUTOREAD. 15 – 0 display mode message; 1 suppress mode message.

Type	Offset SPL	Offset Other	#Words	Name	Content
Logical	56	57	1	ENVIRON	First byte = logical device number of terminal. Second byte = reserved
	57	58	1	USER'TIME*	User-defined time-out length
	58	59	1	IDENTIFIER	Terminal type
	59	60	1	LAB'INFO	First byte = number of labels the terminal supports. Second byte = length of labels (characters).
	60	61	10	Reserved – leave 10 words initialized to zero	
THIS AREA REFERENCED ONLY WHEN USING HP 3075/76 TERMINALS.					
Logical	70	71	1	NUM'FLDS	Number of fields on current form.
1-word integer	71	72	1	SPLIT'PAUSE	Length of time in seconds to pause between lines of text. Default: 3 seconds. –1 = wait for user to hit key. 0 = do not pause >0 = wait specified number of seconds.
	72	73	1	LEFT'MODULE	Which, if any, module is present: 0 = no module; 1 = printer; 2 = multi-function reader; 3 = RS232 interface; 4 = typev badge reader, 5 = magstripe reader, 6 = barcode reader; 7 = HP-IB interface.
	73	74	1	RIGHT'MODULE	Which, if any, module is present: 0 = no module; 1 = printer; 2 = multi-function reader; 3 = RS232 interface; 4 = typev badge reader; 5 = magstripe reader; 6 = barcode reader; 7 = HP-IB interface.
	74	75	1	KEYBOARD	Keyboard type: 0 = standard keyboard (12 function keys); 1 = alphanumeric keyboard (28 function keys).
	75	76	1	DISPLAY	Terminal display type: 0 = numeric display; 1 = alphanumeric display ; 2 = mini-CRT display.
	76	77	1	KEYBOARD'OVER	Whether to override keyboard input. –1 = Override and enable the keyboard without regard to forms design. 0 = Do not override (Default.)
	77	78	1	ERROR'LIGHT	First byte indicates which light will be turned on when an error is detected. (Default: E.) Second byte is reserved.

Type	Offset SPL	Offset Other	#Words	Name	Content
2-word integer	78	79	2	USER'LIGHTS'ON	Indicates whether additional lights will be turned on during run time. 0 = OFF; 1 = ON. (Default: OFF.) Word 1: 1 in bit position 0 = "@" light ON 1 in bit position 1 = "A" light ON 1 in bit position 2 = "B" light ON . . . 1 in bit position 15 = "O" light ON Word 2: 1 in bit position 0 = "P" light ON The remaining bits are reserved.
	80	81	6	Reserved — leave 6 words initialized to zero	

*Not supported on the 3075/76 terminals.

Data Types

Data Type	Data Types for V/3000 Languages			
	COBOL	FORTRAN	BASIC	SPL
1-word integer	COMP PIC S9 thru PIC S9(4)	INTEGER	INTEGER	INTEGER
1-word unsigned integer	COMP PIC 9 thru PIC 9(4)	INTEGER	INTEGER (≤32767)	LOGICAL
2-word integer	COMP PIC S9(5) thru PIC S9(9)	DOUBLE INTEGER	INTEGER INTEGER	DOUBLE INTEGER
character	DISPLAY PIC X(n)	CHARACTER	STRING	BYTE ARRAY

RPG INTERFACE

H SPEC

Columns 7-14	Name of file for runtime error dump; if omitted, dump is sent to terminal

F SPEC

Columns 7-14	Name of file assigned to WORKSTN device
Column 15	U - file must be type update
Column 16	D - file is demand file (usual for HP VIEW interface) P - file is primary file
Column 19	V - record length must be variable
Columns 24-27	Record length; length of data in longest form plus 20 characters for control
Columns 40-46	WORKSTN - required device class name for HP VIEW
Column 51	0-9 number of seconds for message display; default = 3
Column 52	B - enable break key; default is to disable break
Column 53	K - file continuation; 1 for each additional file
Columns 54-59	FORMS - forms file is used (required) BATCH - batch file is used (optional) TRACE - trace file is used (optional)
Columns 60-74	File name of each continuation file

Event Codes

Event codes, indicating expected input, are entered in columns 27 and 33 of I SPEC; 1st digit in 27, 2nd in 33.

Event Code	Function	Response to Action
00	ENTER key pressed at terminal	RDTERM 54
01	f1 key pressed at terminal	RDTERM 54
02	f2 key pressed at terminal	RDTERM 54
03	f3 key pressed at terminal	RDTERM 54
04	f4 key pressed at terminal	RDTERM 54
05	f5 key pressed at terminal	RDTERM 54
06	f6 key pressed at terminal	RDTERM 54
07	f7 key pressed at terminal	RDTERM 54
08	f8 key pressed at terminal	RDTERM 54
09	Read number of fields in error	EDITS 59 NUMERR 61
10	Read data from data buffer	GETDTA 64
11	Read record containing: • record # of batch record • mode (0=collect, 1=browse) • repeatapp or freezeapp status • next form name	(any except 54, 59, 61, 64, or 74)
12	Read length and contents of field from data buffer	GETFLD 74

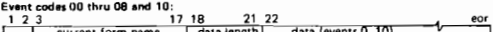
Action Codes

Action codes determine particular action to be taken; code or 6-character action mnemonic are entered in columns 33 through 42 of C SPEC.

Action Code	Mnemonic	Function
50	CHGNXT	Specify next form name; repeat/append; freeze/append
51	GETNXT	Get next form from forms file
52	PUTMSG	Move message to window buffer in memory
53	SHOW	Display current form, initial data, any message
54	RDTERM	Read input from terminal to data buffer
55	SHOMSG	Display message, any new data
56	CDRERR	Display fields with errors, message for 1st error, read user response
57	SHODTA	Display data from user program buffer
58	INIT	Initialize fields in current form
59	EDITS	Perform edits on fields in current form
60	PRINT	Print current form/data on line printer
61	NUMERR	Request error status
62	BADFLD	Request # of field that failed edit and message
63	PUTDTA	Read data from user program to data buffer
64	GETDTA	Write data in data buffer to user program
65	FINISH	Perform final processing on current form
66	WRTBAT	Write data in data buffer to batch file
67	PREV	Read data from previous batch record to data buffer; place program in browse mode
68	REREAD	In browse mode, read data from current batch record into data buffer
69	NEXT	In browse mode, read data from next batch record into data buffer
70	RESUME	Return to collect mode
71	DELETE	In browse mode, delete current batch record
72	RDBTNU	Read batch record identified by its record number
73	CLRMSG CLRMSGI	Clear message from window buffer in memory Clear message from screen and window buffer in memory
74	GETFLD	Locate data from specified field in data buffer
75	PUTFLD	Transfer data from user program to specified field in data buffer

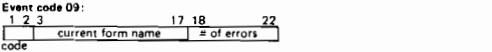
Input Record Formats

Event codes 00 thru 08 and 10:

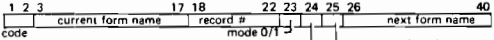


Event code 09:

Event code 11:



Event code 12:

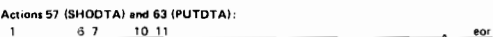


Output Record Formats

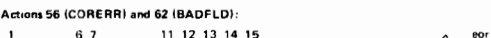
Action 50 (CHGNXT):

Actions 57 (SHODTA) and 63 (PUTDTA):

Actions 56 (CORERR) and 62 (BADFLD):



Actions 52 (PUTMSG) and 55 (SHOMSG):



Actions 74 (GETFLD) and 75 (PUTFLD):

Actions 51 (GETNXT), 53 (SHOW), 54 (RDTERM), 58-61 (INIT,EDITS,PRINT, NUMERR), and 64-73 (GETDTA,FINISH,WRTBAT,PREV,REREAD,NEXT,RESUME, DELETE, RDBTNU, CLRMSG):

ENTRY

:RUN ENTRY.PUB.SYS

Enter Forms File name and press RETURN:

Enter Batch File name and press RETURN:

Forms file is existing file containing form definitions

Batch file is new or existing MPE file name; may be fully qualified

Collect Mode:

f1	f2	f3	f4
HEAD FORM		PRINT	REFRESH
	NEXT FORM	BROWSE	EXIT
f5	f6	f7	f8

Browse Mode:

f1		f3	f4
FIRST REC	OELETE REC	PRINT	REFRESH
PREV REC	NEXT REC	COLLECT	EXIT
f5	f6	f7	f8

Section VI

Utilities

UTILITIES

ASOCTABL

To input from an EDITOR file enter

:FILE INPUT = filename

Default: Input from a terminal.

Then enter Operation

:RUN ASOCTABL.PUB.SYS

Capability: SM or SAVE and WRITE access to PUB.SYS.

Note: Format for input to EDITOR file:

deviceclassname= { @.@ / @.acct / user.acct / user.@ } [, { @.@ / @.acct / user.acct / user.@ } ...]

DISKED2

Operation:

:RUN DISKED2.PUB.SYS

Commands:

>BASE abssector

>DEBUG

>DISC ldev

>DUMP [relsector] [,[numsectors] [,A]]

Note: At least one parameter must be specified with the DUMP command.

>EXIT

>HELP

>LIST [ldev / devclass]

> **MODIFY sectornum, relwordaddr** [,numwords]

>WIDTH

Defaults: relsector=0, numsectors=1, abssectors=0, **$STDLIST**, numwords=1

Capability: SM, PM

Note: Formal file designator - DEDILIST

DPAN4

Operation:

:RUN DPAN4.PUB.SYS [;PARM=10]

Respond with tape number to message:

?!0/time/#{J / S}jsnum/pin/LDEV # FOR "MDUMP" ON TAPE (NUM).

Optional Parameters:

PARM	MEANING
10	Initiate interactive dialogue.

Note: If PARM=10 is included, there will be a short pause before the interactive dialogue is initiated. Respond to the prompts with YES, NO, or ALL.

For more information about DPAN4 and the interactive dialogue, consult the MPE System Utilities Reference Manual (part no. 30000-90044).

Entry Points:

ENTRY POINT	OPERATION
EIGHTLPI	Print entire dump at 8 lines per inch. (Make sure printer is set up for 8 lines per inch.)

DPAN4 output is transmitted to $STDLIST unless run from session then output is to DEV=LP.

If you print a dump on a system which is different from the one on which the dump was taken, make certain that you have a copy of the original (dump system) LOADMAP file. Enter a file command:

```
:FILE LOADMAP.PUB.SYS=dumpload.grp.acct
```

Then run DPAN4.

FREE2

Operation:

:RUN FREE2.PUB.SYS

Note: Formal file designator – FREE2OUT

To redirect the output enter

:FILE FREE2OUT; DEV=LP
:RUN FREE2.PUB.SYS

LISTDIR2

Operation:

:RUN LISTDIR2.PUB.SYS

Commands:

>LISTACCT [accountname] [,listfile] [;PASS]

>LISTGROUP [groupname] [.accountname] [,listfile] [;PASS]

>LISTUSER [username] [.accountname] [,listfile] [;PASS]

>LISTSEC [filename] [.groupname[.accountname]] [,listfile] [;PASS]

>LISTF filename[.groupname[.accountname]] [,listfile] [;PASS] [;MAP]

>HELP

>MOUNT $\left[\left\{\begin{matrix}* \\ \text{vsname}\end{matrix}\right\}\right.$ [.groupname[.acctname]]$\left.\right]$[;GEN=[geninx]]

>DISMOUNT

>EXIT

Note: To redirect the output later
:FILE OUT; DEV=LP
:RUN LISTDIR2.PUB.SYS;PARM=1

No special capabilities are required; however not having certain capabilities, i.e., SM or AM, may restrict the information you can list.

Utilities

LISTLOG2

Operation:

:RUN LISTLOG2.PUB.SYS

Default: LP

Capability: SM

Note: Formal file designator LOGLIST
To redirect the output enter
:FILE LOGLIST;DEV=$STDLIST
:RUN LISTLOG2.PUB.SYS

LISTEQ2

Operation:

:RUN LISTEQ2.PUB.SYS

Default: $STDLIST

Note: Formal file designator - LIST
To redirect the output enter
:FILE LIST;DEV=LP
:RUN LISTEQ2.PUB.SYS;PARM=1

MEMTIMER

Operation:

:RUN MEMTIMER.PUB.SYS;PARM=n

Default: 60 minutes (3600 seconds)

Note: n is any integer greater than zero but less than 65536 used to denote the new logging interval in seconds.

MEMLOGAN

Operation:

:RUN MEMLOGAN.PUB.SYS [;PARM=n]

Capability: SM

Note: n is a one digit code that requests the following options.

n	option
0	MEMLOGAN displays the current contents of MEMLOG but makes no change to this file. (If you omit the n parameter from the :RUN command, this option occurs by default.)
1	MEMLOGAN displays current contents of MEMLOG, deletes all previously recorded errors, and then sets this file to the NO-ERROR state.
2	MEMLOGAN displays the current contents of MEMLOG and then deletes this file from the system. (Because of the security provisions assigned to this file, this is the only way to purge this file from the system.)

To redirect the output enter

```
:FILE OUT;DEV=LP
:RUN MEMLOGAN.PUB.SYS
```

Utilities

PATCH

Operation:

:RUN PATCH.PUB.SYS

Commands:

? D, segment-number, address[,number-of-locations]
(displays a code segment)

?M, segment-number, address[,number-of-locations]
(modifies a code segment)

?DG, DB relative-offset, number of words from offset
(displays the global area of the initial stack)

?MG, DB relative-offset, number of words from offset
(modifies the global area of the initial stack).

SADUTIL

Before you can request any SADUTIL functions, you must coldload and initiate SADUTIL.

To create a coldload tape for Series II/III
:RUN SDUPII.HP32230.SUPPORT

To create a coldload flexible disc/tape for Series 30/33/44
:RUN DUSCOPY.HP32231.SUPPORT

To coldload from a standard magnetic tape

Series II/III

STEP	PROCEDURE
1	Mount the cold-load tape containing SADUTIL on the tape unit configured as Device Reference Table (DRT) Entry No. 6, and press the 0 switch (bottom left of tape unit panel).
2	Place all other tape units off-line, or set them to any unit number but zero (by pressing the 1, 2, or 3 switches on these units).
3	Press the RESET button and the LOAD button on the DRT No. 6 tape unit. The tape winds forward and stops at the load point; the LOAD light turns ON.
4	Press the ON-LINE button on the DRT No. 6 tape unit.
5	Set the system switch register on the system control panel to %003006.
6	Press the ENABLE and LOAD switches on the system control panel simultaneously.
7	Set the system switch register to %000001. This indicates that the first program on the cold-load tape, SADUTIL, is to be loaded.
8	Press the RUN switch. This loads SADUTIL and rewinds the tape.
9	When the tape is re-wound, press the return key on the console keyboard. In response, SADUTIL begins execution, initiating a dialogue with you by way of the console.

Utilities

For Series 30/33/44, cold load and initiate the Diagnostic Utility System (DUS) then enter "SADUTIL"

Functions

Print Functions

PDSK [ldn]
PDTT [ldn]
PFRE [ldn]
PVOL [ldn]
PFIL

Specialized Functions

COND [ldn]
EDIT
SAVE
COPY (Series 33 only)

Utility Functions

OUTM [C/P]
CONF [ldn]
STOP

SLPATCH

Operation:

:RUN SLPATCH.PUB.SYS

Commands:

?[segment-name,] **D, segment-displacement**
[,number-of-words]
(displays the contents of an SL segment)

?[segment-name,] **M,segment-displacement**
[,number-of-words]
(modifies the contents of an SL segment)

Capabilities: PM

SPOOK

Operation

:RUN SPOOK.PUB.SYS

Commands:

ALTER [username[.acctname] / devicefileid[,devicefileid]...][,item[,item[,item]]]

APPEND [username[.acctname] / devicefileid[,devicefileid]...;][range[,filename]]

where range is defined as

[[{ recnumber1 / * / FIRST / LAST } [±offset]] [/ { recnumber2 / * / FIRST / LAST } [±offset]] [,count] / [ALL] / [END]]

Utilities

COPY [username[.acctname] / devicefileid[,devicefileid]...;] [range[,filename]]

where range is defined as

LIST [[{ recnumber1 / * / FIRST / LAST } [±offset]] [/ { recnumber2 / * / FIRST / LAST } [±offset]] [,count] / [ALL]]

DEBUG

EXIT

FIND[@] ["string"]

, [[{ recnumber1 / * / FIRST / LAST } [±offset]] [/ { recnumber2 / * / FIRST / LAST } [±offset]] [,count] / [ALL]]

HELP [{ MPE / HELP / tablecontents / command [,keyword] / ALL }]

INPUT [username[.acctname] / devicefileid[,devicefileid]...] ;tapefile

KILL

LIST [[[{ recnumber1 / * / FIRST / LAST } [±offset]] [[,count] [/ { recnumber2 / * / FIRST / LAST } [±offset]]] / {ALL}]

MODE [{ WIDTH= { ±nnn / OFF } / CONTROLS= { ON / OFF } } [{ WIDTH= { ±nnn / OFF } / CONTROLS= { ON / OFF } }] ...]

OUTPUT [username[.acctname] / devicefileid[,devicefileid] ...] ;tapefile[;PURGE]

PURGE devicefileid [,devicefileid] ...

QUIT

RUN [progfile / *]

SHOW [[username[.acctname]] [; [@] [[I / O]]] / devicefileid [,devicefileid] ...]

TEXT devicefileid

VINIT

:**VINIT** [listfile]

Operation

Commands:

>**INIT** [uname, ldn [,vsname.groupname.acctname] [;GEN=gen index]

>**FORMAT** ldn

>**SERIAL** ldn

>SCRATCH ldn[;RESET]

>COPY fromldn, toldn[;GEN=genindex]]

>DSTAT [ldn | ALL or @]

>FOREIGN ldn

>PDEFN [* .group.account | vsname]

>PLABEL ldn

>PFSPACE ldn

>PDTRACK ldn

>COND ldn [;SIZE=n | ;ALL | ;RECOVER]

>DTRACK ldn

>EXIT

>HELP

Capability: SM or OP

XPLAIN

Section VII

Segmenter

Operation

:SEGMENTER [listfile]

Commands:

-ADDRL	rbmname[(index)]
-ADDSL	segname[,PMAP]
-AUXUSL	filereference
-BUILDRL	filereference,records,extents
-BUILDSL	SL[.group],records,extents
-BUILDUSL	filereference,records,extents
-CEASE	[ENTRY, / UNIT, / SEGMENT,] name[(index)]
-CLEANSL	[filename]
-CLEANUSL	[filename]
-COPY	[UNIT, / SEGMENT,] name[(index)]
-COPYSL	percent[,filename]
-COPYUSL	percent[,filename]
-EXIT	
-HIDE	entryname [(index)]
-LISTRL	
-LISTSL	
-LISTUSL	
-NEWSEG	newsegname,rbmname[(index)]
-PREPARE	progfile [;ZERODB] [;PMAP] [;MAXDATA=segsize] [;STACK=stacksize] [;DL=dlsize] [;CAP=caplist] [;RL=filename]

Note: For parameter definitions, see :PREP

-PURGERBM	[UNIT, / SEGMENT,] name[(index)]

Segmenter

-PURGERL [ENTRY, / UNIT,] name

-PURGESL [ENTRY, / SEGMENT,] name

-REVEAL entryname [(index)]

-RL filereference

-SL SL[.group].account]]

-USE [ENTRY, / UNIT, / SEGMENT,] name[(index)]

-USL filereference

Notes:
1. (index) – default is most recent active entry (index=0); most recent=1, oldest=n.
2. records, extents -1024,8 is system default; try those values.
3. BUILDRL, BUILDUSL, or BUILDSL imply RL, USL, or SL command.
4. CEASE and USE activate and deactivate entry points.
5. HIDE and REVEAL go together; HIDE adds OPTION INTERNAL to unit.
6. COPY is from AUXUSL to USL.
7. NEWSEG changes the segment name associated with an rbm in USL.
8. See PMAP and LMAP formats in DEBUG section.

Intrinsics:

ADJUSTUSLF

```
      I                          IV    IV
errnum := ADJUSTUSLF (uslfnum,records);
```

EXPANDUSLF

```
      I                          IV    IV
filenum := EXPANDUSLF (uslfnum,records);
```

INITUSLF

```
      I                      IV    IA
errnum:=INITUSLF(uslfnum,rec0)
```

Section VIII

Intrinsics

```
                IV  LV    O-V
ACTIVATE        (pin, susp);
```

Condition Codes: CCE, CCG, CCL

```
                I                          IV      IV
ADJUSTUSLF      errnum: = ADJUSTUSLF (uslfnum, records);
```

Condition Codes: CCE, CCL

```
                LV  IV  I
ALTDSEG         (index, inc, size);
```

Condition Codes: CCE, CCG, CCL

```
                LV
ARITRAP         (state);
```

Condition Codes: CCE, CCG

```
                I                  IV    IV   BA
ASCII           numchar: = ASCII (word, base, string);
```

Condition Codes: None

```
                L                        BA    IV
BINARY          bineqv: = BINARY (string, length);
```

Condition Codes: CCE, CCG, CCL

```
                L
CALENDAR        date: = CALENDAR;
```

Bits[0:7] : YEAR (e.g., 76)
Bits[7:9] : DAY OF YEAR

Condition Codes: None

```
CAUSEBREAK;
```

Condition Codes: CCE, CCL

```
                I                          IV      BA
CLEANUSL        filenum: = CLEANUSL (uslfnum, filename);
```

Condition Codes: CCE, CCL

Note: Requires 3000 words of available stack space to execute.

CLOCK

```
 D
time: = CLOCK;
```

Word 1 Bits[0:8] Hour
Bits[8:8] Minute
Word 2 Bits[0:8] Second
Bits[8:8] 10th of Second

Condition Codes: None

```
                 D       I       I
CLOSELOG      (index, mode, status);
```

Condition Codes: None

```
                  BA        I       I
COMMAND       (comimage, error, parm);
```

Condition Codes: CCE, CCG, CCL

Note: Comimage≡MPE command with no ':' ends with %15(CR).

Error: =(0≡no error | n≡command Interpreter Error Code).

PARM: =Index of erroneous parameter (CCG only)

```
                  BA        BA        I     IV     LV
CREATE        (progname, entryname, pin, param, flags,

                 IV       IV       IV        LV        IV     O-V
              stacksize, dlsize, maxdata, priorityclass, rank);
```

Condition Codes: CCE, CCG, CCL

```
                    I     I      BA        IA      LA      O-V
CREATEPROCESS   (error,pin,progname,itemnums,items);
```

Condition Codes: CCE,CCG,CCL

```
                    IV    BA      BA          IV        BA     O-V
CTRANSLATE   (code, instring, outstring, stringlength, table);
```

Condition Codes: CCE, CCL

Note; Translates characters from ASCII to EBCDIC or another code specified in **table** or vise-versa.

Code: =(≡use table | 1≡EBCDIC to ASCII | 2≡ASCII to EBCDIC)

```
                    I                    DV    IV    BA
DASCII        numchar: =DASCII (dword, base, string);
```

Condition Codes: None

```
                BA
              (datebuf);

DATELINE      <<Buf = 27 bytes: = "Fri,  May  25,  1979,
                 12:06 PM">>
```

Note: Spaces must be used exactly as shown on example.

Condition Codes: None

```
                D                    BA     IV
DBINARY       dval: = DBINARY (string, length);
```

Condition Codes: CCE, CCG, CCL

```
DEBUG;
```

Condition Codes: None

```
                I                    IV
DLSIZE        dldbsize: = DLSIZE (size);
```

Condition Codes: CCE, CCG, CCL

```
                LV    IV    IV      LA
DMOVIN        (index, disp, number, location);
```

Condition Codes: CCE, CCG, CCL

```
                 LV    IV    IV      LA
DMOVOUT        (index, disp, number, location);
```

Condition Codes: CCE, CCG, CCL

```
                 I                                IV      IV
EXPANDUSLF   filenum: = EXPANDUSLF (uslfnum, records);
```

Condition Codes: CCE, CCL

```
                 I
FATHER         pin: = FATHER;
```

Condition Codes: CCE, CCG, CCL

```
                  I      I      IA      I      I
FCARD          (recode, filenum, bufadr, count, status);
```

Condition Codes: None

```
                 IV      I       I      D       I      O-V
FCHECK         (filenum, errorcode, tlog, blknum, numrecs);
```

<<file errors>>

Condition Codes: CCE, CCL

```
                 IV       IV       IV
FCLOSE         (filenum, disposition, seccode);
```

Condition Codes: CCE, CCL

Note: Disp (13:3): =(0≡no change | 1≡save perm | 2≡save temp | 3≡temp, no rewind | 4≡delete)
Disp (12:1):=(1≡return space beyond EOF | 0≡retain space)
Security Code: =(0≡unrestricted | 1≡restricted)

```
                    IV          IV          L
FCONTROL    (filenum, controlcode, param);
```

Values for controlcode:

0 = General.
1 = LP.
2 = Complete all I/O.
3 = Param:=status.
4 = Set timeout.
5 = Rewind.
6 = Write EOF.
7 = Forward to tape mark.
8 = Back to tape mark.
9 = Rewind/offline.
10 = Change terminal input speed.
11 = Change terminal output speed.
12 = Turn echo facility on.
13 = Turn echo facility off.
14 = Disable the system break function.
15 = Enable the system break function.
16 = Disable the subsystem break function.
17 = Enable the subsystem break function.
18 = Disable tape mode option.
19 = Enable tape mode option.
20 = Disable the terminal input time.
21 = Enable the terminal input timer.
22 = Read the terminal input timer.
23 = Disable parity checking.
24 = Enable parity checking.
25 = Define line-termination characters for terminal input.
26 = Disable binary transfers.
27 = Enable binary transfers.
28 = Disable user block mode transfers.
29 = Enable user block mode transfers.
34 = Disable line deletion echo suppression.
35 = Enable line deletion echo suppression.
36 = Set parity.
37 = Allocate a terminal.
38 = Set terminal type.
39 = Obtain terminal type information.
40 = Obtain terminal output speed.
41 = Set unedited terminal mode.
43 = Abort pending NO-WAIT I/O request.
45 = Enable/Disable extended wait.
46 = Enable/Disable reading writer's ID.
47 = Nondestructive read.

Condition Codes: CCE, CCL

```
                          IV        DV      O-V
FDELETE                 (filenum, recnum);
        Condition Codes: CCE, CCG, CCL

                          IV      LA      IV
FDEVICECONTROL          (filenum, target, tcount,

                          LV        LV      LV        I
                        controlcode, parm1, parm2, errnum);
```

CONTROLCODE	FUNCTION
128	Select primary/secondary character set
129	Select logical pages/forms
130	Move pen relative
131	Move pen absolute
132	Define job characteristics
133	Download physical page definition
134	Download/delete character set
135	Download/delete forms
136	Download logical page table
137	Download multi-copy form overlay table
138	Download/delete VFC

ERRNUM	MEANING
126	Character set number out of range
127	Form number out of range
128	Logical page number out of range
129	VFC number out of range
130	Number of copies out of range
131	Tcount parameter incorrect
132	Form identifier number out of range

```
                    I        LA        I
FERRMSG           (errorcode, msgbuf, msglgth);
        Condition Codes: CCE, CCL, CCG

                    IV        IV          8A
FFILEINFO         (filenum [,itemnum1, itemvalue1]
                           [,itemnum2, itemvalue2]
                           [,itemnum3, itemvalue3]
                           [,itemnum4, itemvalue4]
                           [,itemnum5, itemvalue5]);
```

ITEM NO.	TYPE	ITEM	UNITS
1	BA	filename (see FGETINFO)	
2	L	foptions (see FGETINFO)	
3	L	aoptions (see FGETINFO)	
4	I	recsize (see FGETINFO)	words/bytes
5	I	devtype (see FGETINFO)	
6	L	ldnum (see FGETINFO)	
7	L	hdaddr (see FGETINFO)	
8	I	filecode (see FGETINFO)	
9	D	recpt (see FGETINFO)	
10	D	eof (see FGETINFO)	
11	D	flimit (see FGETINFO)	records
12	D	logcount (see FGETINFO)	records
13	D	physcount (see FGETINFO)	records
14	I	blksize (see FGETINFO)	words/bytes
15	L	extsize (see FGETINFO)	sectors
16	I	numextents (see FGETINFO)	
17	I	userlabels (see FBETINFO)	
18	BA	creatorid (see FGETINFO)	
19	D	labaddr (see FGETINFO)	
20	I	blocking factor (See FOPEN)	
21	I	physical block size	words
22	I	data block size	words
23	I	offset to data in blocks	words
24	I	offset to Active Record Table in block (R10 files)	words
25	I	size of Active Record Table	words
26	BA	vol. ID (label tape) (see Label Tapes)	
27	BA	vol. set ID (label tape) (see Label Tapes)	
28	I	expiration date (CALENDAR format (see Label Tapes)	
29	I	file sequence number (see Label Tapes)	
30	I	reel number (see Label Tapes)	
31	I	sequence type (see Label Tapes)	
32	I	creation date (CALENDAR Format (see Label Tapes)	
33	I	label type (see Label Tapes)	
34		RESERVED	
35			

36	L	File Allocation Date (CALENDAR format)
37	D	File Allocation Time (CLOCK format)
38	L	SPOOFLE Device File Number (#O or #I number) (see File Code)
40	D	disc or diskette device status
41	I	device type
42	I	device subtype
43	BA	Spoolfile environment
44	I	[Reserved for system use]
45	BA	File name of labeled tape
46	I	Density of tape file (valid only for files on an HP 7976A tape drive)

Note: Parameters must appear in pairs.

Condition Codes: CCE, CCL

```
                IV      BA        L        L        I
FGETINFO   (filenum, filename, foptions, aoptions, recsize,

                I        L        L        I        D   D
           devtype, ldnum, hdaddr, filecode, recpt, eof,

                D        D        D        I        L
           flimit, logcount, physcount, blksize, extsize,

                I        I        BA       D       O-V
           numextents, userlabels, creatorid, labaddr);
```

Note: filename parm must be 28 bytes long.

Condition Codes: CCE, CCL

```
                BA        L        I
FINDJCW    (jcwname, jcwvalue, status);
```

Condition Codes: None
(status <> 0 is error)

```
                IV       LV
FLOCK      (filenum, lockcond);
```

Condition Codes: CCE, CCG, CCL

```
                    LV    BA
FMTCALENDAR (date, string);
```

Condition Codes: None

```
                DV    BA
FMTCLOCK (time, string);
```

Condition Codes: None

```
                    LV   DV    BA
FMTDATE        (date, time, string);
```

Condition Codes: None

FOPEN

```
   I                          BA          LV        LV       IV
filenum: = FOPEN (formaldesignator, foptions, aoptions, recsize,

          BA       BA        IV          IV            IV
        device, formmsg, userlabels, blockfactor, numbuffers,

          DV       IV          IV        IV       O-V
        filesize, numextents, initialloc, filecode);
```

<< returns "Filenum' used in other intrinsics>>

Condition Codes: CCE, CCL

```
                    IV       DV
FPOINT         (filenum, recnum);
```

<<point to a record directly>>

Condition Codes: CCE, CCG, CCL

```
                I                IV      LA      IV
FREAD          lgth: = FREAD (filenum, target, tcount);
```

Condition Codes: CCE, CCG, CCL

```
                    I                                IV
FREADBACKWARD   lgth: = FREADBACKWARD (filenum,
                                LA      IV
                            target, tcount);
```

Condition Codes: CCE, CCG, CCL

FOPTIONS:

BITS	(0:2)	(2:3)		(5:1)	(6:1)	(7:1)	(8:2)	(10:3)	(13:1)	(14:2)
FIELD	Reserved	File Type		Disallow :FILE	MPE Tape Labels	Carriage Control	Record Format	Default Designator	ASCII/ Binary	Domain
MEANING		00	0=STD	0=Allow :FILE	0=NON LABEL-ED TAPE	0=NOCCTL	00=Fixed	000=filename	0=Binary	00=New file
		00	1=KSAM	1=No :FILE	1= LABEL-ED TAPE	1=CCTL	01=Variable	001=$STDLIST	1=ASCII	01=Old Permanent File
		01	0=RIO				10=Unde-fined	010=$NEWPASS		
		10	0=CIR					011=$OLDPASS		10=Old Temporary File
		11	0=MSG					100=$STDIN		
								101=$STDINX		11=Old Perm. or Temp. File
								110=$NULL		

BITS	(0:3)	(3:1)	(4:1)	(5:2)	(7:1)	(8:2)	(10:1)	(11:1)	(12:4)	
FIELD	Reserved	File Copy	No-Wait I/O	Multi Access	Inhibit Buffering	Exclusive Access	Dynamic Locking	Multi-record Access	Access Type	
MEANING		0=access in file's native mode 1=access as standard sequential file	1=No Wait 2=Non No-Wait	00=Non-multi-access 01=Only Intra-job multi-access 10=Inter-job multi-access allowed	0=BUF 1=NOBUF	00=Default 01=Exclusive 10=Exclusive Access Read 11=Share	0=No FLOCK Allowed 1: FLOCK Allowed	0=No Multi-Record 1=Multi record	0	000=Read only
									0	001=Write only
									0	010=Write (save) only
									0	011=Append only
									0	100=Read/write
									0	101=Update
									0	110=Execute

```
                IV      LA      IV      DV
FREADDIR        (filenum, target, tcount, recnum);
```

Condition Codes: CCE, CCG, CCL

```
                IV      LA      IV      IV      O-V
FREADLABEL      (filenum, target, tcount, labelid);
```

Condition Codes: CCE, CCG, CCL

```
                IV      DV
FREADSEEK       (filenum, recnum);
```

Condition Codes: CCE, CCG, CCL

```
                LV  LV
FREEDSEG        (index, id);
```

Condition Codes: CCE, CCG, CCL

```
FREELOCRIN;
```

Condition Codes: CCE, CCG, CCL

```
                L                               IV      IV
FRELATE         intordup: = FRELATE (infilenum, listfilenum);
```

Condition Codes: CCE, CCG, CCL

```
                IV              BA
FRENAME         (filenum, newfilereference);
```

Condition Codes: CCE, CCL

```
                IV              LV
FSETMODE        (filenum, modeflags);
```

Note: Modeflags = (14:1) ← (0: = unblocked I/O, w - o wait | 1≡complete are I/O)
(13:1) ← (0: = automatic CR/LF | 1≡suppress CR/LF)
(12:1) ← (0: = CCE for tape errors | 1≡CCL for tape errors)

Condition Codes: CCE, CCL

```
                    IV          IV
FSPACE          (filenum, displacement);
```

<<+≡forward, -≡backward>>

Condition Codes: CCE, CCG, CCL

```
                    IV
FUNLOCK         (filenum);
```

Condition Codes: CCE, CCG, CCL

```
                    IV      LA      IV
FUPDATE         (filenum, target, tcount);
```

Condition Codes: CCE, CCG, CCL

```
                    IV      LA      IV      LV
FWRITE          (filenum, target, tcount, control);
```

Note: Control: =(0≡normal | 1≡ use 1st character of "target" | "1"≡ page eject | %320 ≡ no cr, no LF)

Condition Codes: CCE, CCG, CCL

```
                    IV      LA      IV      DV
FWRITEDIR       (filenum, target, tcount, recnum);
```

Condition Codes: CCE, CCG, CCL

```
                    IV      LA      IV      IV      O-V
FWRITELABEL     (filenum, target, tcount, labelid);
```

Condition Codes: CCE, CCG, CCL

GENMESSAGE

```
  I                                IV      IV      IV      BA      IV
msglen: = GENMESSAGE (filenum, setnum, msgnum, buff, buffsize,

          LV      LV      LV      LV      LV      LV      IV
        parmask, parm1, parm2, parm3, parm4, parm5, msgdest,

          I      O-V
        errnum);
```

Condition Codes: CCE, CCG, CCL

```
                  I              IA I      I
GET             ifun: = GET(itag,il,ionumber)

                  L      I     LV
GETDSEG         (index, length, id);
```

Condition Codes: CCE, CCG, CCL

```
                 L
GETJCW          jcw: =GETJCW;
```

Condition Codes: None

```
                   LV
GETLOCRIN       (rincount);
```

Condition Codes: CCE, CCG, CCL

```
                  I
GETORIGIN       source: = GETORIGIN;
```

Condition Codes: None

```
                 IV      LV        IV    O-V
GETPRIORITY     (pin, priorityclass, rank);
```

Condition Codes: CCE, CCG, CCL

```
GETPRIVMODE; O-P
```

Condition Codes: CCE, CCG

```
                 I                      IV
GETPROCID       pin: = GETPROCID (numson);
```

Condition Codes: None

```
                  D                          IV
GETPROCINFO     statinfo: = GETPROCINFO (pin);
```

Condition Codes: CCE, CCG, CCL

GETUSERMODE;

Condition Codes: CCE, CCG

```
                    I                          IV      IA
INITUSLF        errnum: = INITUSLF (uslfnum, rec0);
```

Condition Codes: CCE, CCL

IODONTWAIT

```
 I                          IV      LA      I        L      O-V
fnum: = IODONTWAIT (filenum, target, tcount, cstation);
```

Condition Codes: CCE, CCG, CCL

IOWAIT

```
 I                  IV      LA      I        L      O-V
fnum: = IOWAIT (filenum, target, tcount, cstation);
```

Condition Codes: CCE, CCG, CCL

```
                  IV
KILL            (pin);
```

Condition Codes: CCE, CCG, CCL

```
                    I                          BA      IV      I
LOADPROC        identnum: =LOADPROC (procname, lib, plabel);
```

Note: Library: = (0≡SYS SL | 1≡Acct, SYS SL | 2≡Grp, Acct, SYS SL)

Condition Codes: CCE, CCL

```
                    IV        L          BA
LOCKGLORIN   (rinnum, lockcond, rinpassword);
```

Condition Codes: If lockcond = TRUE:
CCE = Request Granted
CCL = Request denied

If lockcond = FALSE
CCE = Request granted
CCG = Request denied
CCL = Request denied

Note: lockcond may be changed by intrinsic on return.

```
                    IV        L
LOCKLOCRIN (rinnum, lockcond);
```

Condition Codes: If lockcond = TRUE:
CCE, CCL

If lockcond = FALSE:
CCE, CCL, CCG

```
                    I                          IV
LOCRINOWNER pin: = LOCRINOWNER (rinnum);
```

Condition Codes: CCE, CCG, CCL

```
                    L              IV     I
MAIL            status: = MAIL (pin, count);
```

Condition Codes: CCE, CCG

```
MYCOMMAND
   I                                BA          BA          IV
entryno: = MYCOMMAND (comimage, delimiters, maxparms,
                I        DA    BA    BP   O-V
          numparms, parms, dict, defn);
```

Condition Codes: CCE, CCG, CCL

```
                    D     LA    LA    I       I
OPENLOG         (index, logid, pass, mode, status);
```

Condition Codes: None

```
                    R
PAUSE           (interval);
```

Note: interval specifies the amount of time in seconds.

Condition Codes: CCE, CCG, CCL

```
PCHECK

   I                    IV
icode: = PCHECK (dsnum);
                    IV
PCLOSE          (dsnum);
                    IV     IA
PCONTROL        (dsnum,itag);
```

```
POPEN
  I                    BA      BA      IA      BA      IV    LV
dsnum: = POPEN (dsdevice,progname,itag,entryname,param,flags,
                    IV      IV      IV      IV
                stacksize,dlsize,maxdata,bufsize);
```

```
PREAD
 I                  IV    IA    IV    IA
lgth: = PREAD (dsnum,target,tcount,itag);
```

```
                    LA      IV      IV
PRINT           (message, length, control);
```

Condition Codes: CCE, CCG, CCL

```
                    IV
PRINTFILEINFO (fnum);
```

Condition Codes: None

```
                    LA      IV      IV
PRINTOP         (message, length, control);
```

Note: Message limit – 56 characters

Condition Codes: CCE, CCL

```
PRINTOPREPLY
 I                          LA      IV      IV      LA      IV
lgth: = PRINTOPREPLY (message, length, control, reply, expectedl);
```

Condition Codes: CCE, CCL

```
PROCTIME
  D
Time: = PROCTIME;
```

Condition Codes: None

```
                    IV      IV
PTAPE           (filenum1, filenum2);
```

Condition Codes: CCE, CCG, CCL

```
                    BA      L       I
PUTJCW          (jcwname, jcwvalue, status);
```

Condition Codes: None
(status <> 0 is error)

```
                    IV   IA   IV   IA
PWRITE          (dsnum,target,tcount,itag);

                    IV
QUIT            (num);
```

Condition Codes: None

```
                    IV
QUITPROG          (num);
```

Condition Codes: None

```
                    I                  LA        IV
READ              lgth: = READ (message, expectedl);
```

Condition Codes: CCE, CCG, CCL

```
                    I                  LA        IV
READX             lgth: = READX (message, expectedl);
```

Condition Codes: CCE, CCG, CCL

```
                    L                        IV    LA       LV
RECEIVEMAIL       status: = RECEIVEMAIL (pin, location, waitflag);
```

Condition Codes: CCE, CCG, CCL

```
                    IA
REJECT            (itag);
```

RESETCONTROL;

Condition Codes: CCE, CCL

RESETDUMP;

Condition Codes: CCE, CCG

SEARCH

```
  I                    BA     IV    BA   BP   O-V
entryno: = SEARCH (target, length, dict, defn);
```

Condition Codes: None

SENDMAIL

```
  L                    IV    IV     LA        LV
status: = SENDMAIL (pin, count, location, waitflag);
```

Condition Codes: CCE, CCG, CCL

```
                    LV
SETDUMP           (flags);
```

Condition Codes: CCE, CCG

```
                    LV
SETJCW          (word);
```

Condition Codes: None

```
                    BA        I        L    DA    O-V
STACKDUMP    (filename, idnumber, flags, selec);
```

Condition Codes: CCE, CCG, CCL

```
                  LV  IV   O-V
SUSPEND         (susp, rin);
```

Condition Codes: CCE, CCL

```
                    L                          LV    O-P
SWITCHDB        logindex: = SWITCHDB (index);
```

Note: Requires privileged mode

Condition Codes: CCE, CCL

```
TERMINATE;
```

Condition Codes: None

```
                    D
TIMER           count: = TIMER;
```

Condition Codes: None

```
                    IV
UNLOADPROC   (procid);
```

Condition Codes: CCE, CCL

```
                    IV
UNLOCKGLORIN  (rinnum);
```

Condition Codes: CCE, CCG, CCL

```
                    IV
UNLOCKLOCRIN  (rinnum);
```

Condition Codes: CCE, CCG, CCL

WHO (mode, capability, lattr, usern, groupn, acctn, homen, termn);

L D D BA BA BA BA L O-V

Note: Mode (15:1):=(0≡not interactive | 1≡interactive)
(14:1):=(0≡not duplicative | 1≡duplicative)
(12:1):=(1≡SESSION | 2≡JOB)

Condition Codes: None

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
WORO 1	S	A	A	G	D	O	C	U	L	X	X	X	X	C	N	S
	M	M	L	L	I	P	V	V	G	X	X	X	X	S	D	F
WORO 2	X	X	X	X	X	X	X	B	I	P	X	X	M	X	D	P
	X	X	X	X	X	X	X	A	A	M	X	X	R	X	S	H

X
X = Not used

Condition Codes: None

WRITELOG (index, data, len, mode, status);

D LA I I I

Condition Codes: None

XARITRAP (mask, plabel, oldmask, oldplabel);

IV IV I I

Condition Codes: CCE, CCG, CCL

XCONTRAP (plabel, oldplabel);

IV I

Condition Codes: CCE, CCG, CCL

```
                    IV      I
XLIBTRAP      (plabel, oldplabel);
```

Condition Codes: CCE, CCG, CCL

```
                    IV      I
XSYSTRAP      (plabel, oldplabel);
```

Condition Codes: CCE, CCG, CCL

```
                     I              IV
ZSIZE          actsize: =ZSIZE (size);
```

Condition Codes: CCE, CCG, CCL

Section IX

Debug

DEBUG

Operation

1. Direct call to DEBUG intrinsic.
2. Reaching breakpoint.
3. ;DEBUG parameter in :PREPRUN or :RUN command (sets breakpoint on first executable instruction), or
4. DEBUG Bit in FLAG. (13:1) parameter of CREATE intrinsic.
5. :DEBUG command (user must have PM capability)

Access Scope

Determined by user capability (Privileged vs. Non-Privileged). NOT ACCESSIBLE FROM BATCH MODE – calls treated as NOP.

Messages

Call/Breakpoint

•{DEBUG / BREAK}• |PRIV| <LOCATION>

Error

SYNTAX	n	INVALID SYNTAX
NO-NO	n	INVALID INFO WAS INPUT
BOUNDS	n	BOUNDS VIOLATION
FULL	n	BREAKPOINT TABLE FULL
SAME	n	SYSTEM/PRIVATE BREAKPOINT CONFLICT EXISTING
CHECK	n	NEW BREAKPOINT CONFLICTS WITH EXISTING BREAKPOINT
	n	Byte index within command string which is in error.

Debug

Command Syntax

1. Prompt is ?

2. Boxed information applies to Privileged Mode (PM) only.

3. An expression (*expr*) may be substituted where any numeric field is allowed.

4. Octal values (single precision) indicated by optional %.

5. In commands, *segmt.* refers to *logical segment number* obtained from PMAP for user program files (see format on page 115) or from last column of LMAP for Segmented Library Procedures (see page 116).

Command Operation

1. Bounds checking performed in user mode and when setting breakpoints.

2. Instruction where breakpoint occurred is not executed prior to entering Debug.

3. All numbers, address, etc. default to octal.

4. Logical arithmetic used throughout.

5. In privileged mode an improper *ldev* (logical device number) parameter supplied in a command may destroy the system.

6. Privileged users may access absolute addresses, systems data, etc.

7. DEBUG parameter of *:RUN progfile* command is ignored if user does not have read (R) and write (W) access to *progfile.* (See File Security, page 95.)

8. All numbers are one-word values, assumed as octal. (Double-word integers are not allowed.)

Breakpoint Commands

	Privileged Mode
AP	Allow private breakpoints.
AS	Allow system breakpoints.

B[segmt.] offset, . . .	Set temporary breakpoint (break first time only).
B[segmt.] offset:n, . . .	Set temporary breakpoint; break once on the nth execution. (DO NOT USE ON BRANCH.)
B[segmt.] offset:cond, . . .	Set temporary breakpoint; break once when cond is true.
B[segmt.] offset:@, . . .	Set permanent breakpoint, break always. (DO NOT USE ON BRANCH.)
B[segmt.] offset:@n, . . .	Set permanent breakpoint; break on every nth execution. (DO NOT USE ON BRANCH.)
B[segmt.] offset:@cond, . . .	Set permanent breakpoint; break every time cond is true. (DO NOT USE ON BRANCH.)

Note:
For group and account breakpoints, use prefix G (for group) or P (for account) before segment identifier:

$\begin{bmatrix} G \\ P \end{bmatrix}$ [segment.] offset . . .

B@	List all breakpoints.

Privileged Mode

BS[segment.] offset. . .	Debug system SL.
BA[segmt.] offset, . . .	Set breakpoint in actual CST. Actual CST numbers are obtained from LMAP or from MPE system load map (printed at system start-up time).

To debug another process, use pin numbers:

B [pin. [G S P A] segmt.] offset [:[@] [n cond]] , . . .

B[pin.]@	List all breakpoints in specified process.

Note:
The listing format for each breakpoint is

LCST=[P/G] lsn, P=pc, CST=asn, [@] t/u

where:

P
Account Public SL.

G
Group SL.

lsn
Logical CST no.

pc
Program counter.

asn
Actual CST no.

@
Permanent breakpoint. (No @ indicates temporary breakpoint.)

t
Total number of executions allowed by conditional breakpoint.

u
Total number of times breakpoint actually executed.

C [[G/P] segment] offset,...

C@ Clear all user breakpoints.

Privileged Mode

C [[S / A] segment.] offset , . . .

where S indicates breakpoints in the system segment library

A indicates breakpoints in absolute code segment (CST).

To clear breakpoints in another process, use pin number:

C [pin. [G / P / S / A] segment.] offset, . . .

C@ clears all system-owned breakpoints if operating in AS mode or owned breakpoints if operating in AP mode.

C[pin.]@ clears all breakpoints in the specified process.

E[expr] Execute EXIT n (where n = *expr*).

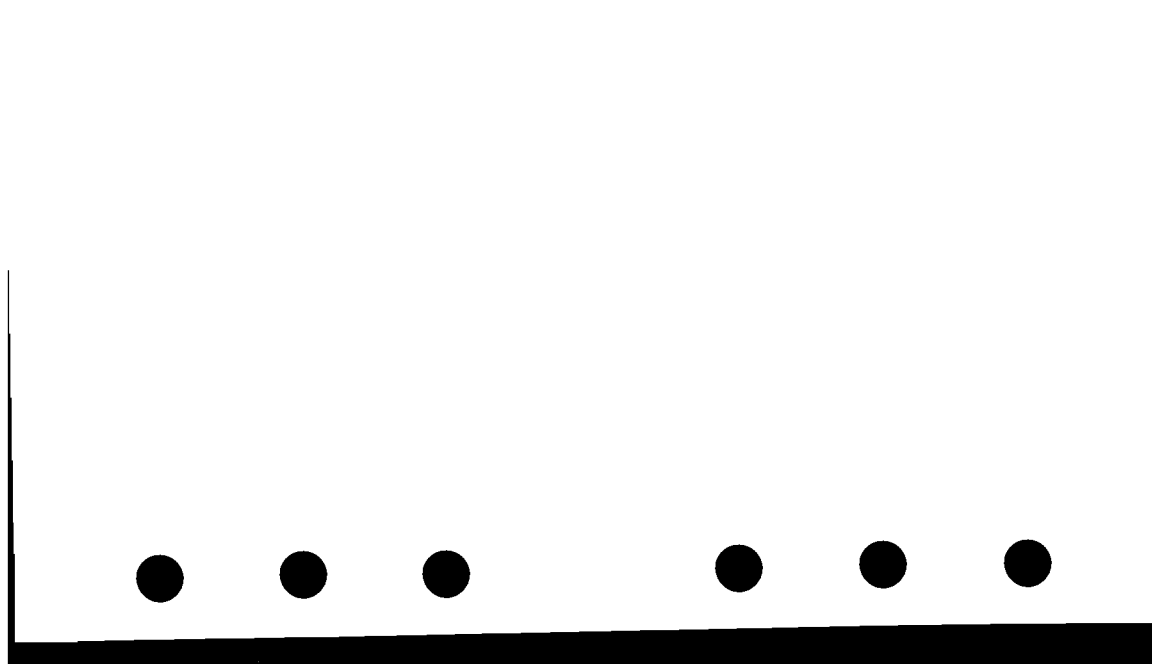
E@ Terminate program.

R[[segmt.] offset [:[@] count / cond]] Resume execution and optionally set another breakpoint.

NOTE

To resume and set a group or account breakpoint, use prefix G for group or P for account before the segment parameter, as follows:

R[[{ G / P } segment.] offset[:[@] count]]

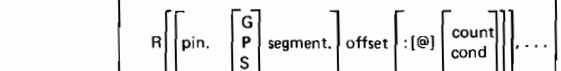


Privileged Mode

To resume and set a breakpoint for system segmented library, use prefix S before the segment parameter, as follows:

R[[S segment.] offset [:[@] count]]

To resume and set a breakpoint in another process, use pin number:



Display/Listing Commands

D[dispbase] [offset] [,count] [,mode] — Display memory.

Note:
dispbase = DB, DL, Q, S, PB, PL, P. (If omitted, DB is assumed.)

offset = Location at which display begins. (: specifies preceding expr. is *indirect* address, as in D Q+5:,#8,A.)

count = No. of locations to display.

mode = O (Octal); I (Decimal Integer); H (hexadecimal); C (code); A (ASCII). Octal is default.

Privileged Mode

A = Absolute Relative (base = location 0).

SY = System Global Relative (base = system base).

CO = Code Segment Relative (base = base of segment).

DA = Data Segment Relative (base = base of segment).

DX = Current Absolute DB Relative (base = absolute DB).

EA = Extended Absolute Address (base = bank specified).

The bank number in EA mode follows EA; for example:

D EA2+10 Displays one word at location 10 of bank 2.

For CO and DA, the offset immediately follows the mnemonic (CO or DA) unless it is an expression involving a calculation when it is enclosed in parentheses; for example:

D DA22+6,6 Displays 6 words starting at location 6 data segment 22.

D CO(4+6),3 Displays first 3 words of segment 12 (octal).

DR[,reg],... Display registers.

Note:
reg = DL, Q, S, Z, X, ST, P, 1, 2, 3, and 4 registers.

Privileged Mode

Privileged mode, if displaying all registers (register parameter omitted), also includes the following values:

PCB = Process Control Block Index.

CST = Absolute Code Segment Index.

STAK = Stack Segment Index.

DST = Extra Data Segment Index.

DX = Current value of DB register, if in absolute mode.

EA = Current bank number, if in absolute mode.

In addition, the segment number displayed as LCST is preceded by S for a system library segment.

Privileged Mode

DV [ldev] Display virtual memory
+ startsector [,count]
[,mode]

Notes:
ldev = logical device no.

count = No. of sectors displayed.

mode = O (Octal)
I (Decimal Integer)
A (ASCII)
Octal is default.

Startsector signifies the starting sector address to be displayed. If the sector address requires more than 16 bits, it must be entered as:

low-order bits: high-order bits

L [fileref] Direct list output to *fileref* file. (If no *fileref*, switch back to user's terminal.)

L0	Closes an open file and switches back to terminal.

Privileged Mode

L [ldev]	Direct list output to *ldev* device. (If no *ldev* switch back to user's terminal.)

Memory/Register Modification Commands

M[modbase] [offset] [,count[,mode]] — Modify memory.

Note:

modbase = DB, DL, Q, S

offset = Location at which modification begins. The : specifies preceding expr. is *indirect* address as in M Q-5:+15,2

count = No. of locations to modify.

mode = O(Octal)
I (Decimal Integer)
A (ASCII)
C (Code)

(CR) Leaves value unchanged.

Terminates command.

Privileged Mode

Other values allowed for modbase:

A = Absolute 0.

SY = System Global.

DA (dst) = Data Segment no. *dst*.

DX = Absolute DB.

EA = Extended Absolute Address.

MR[,register] . . . — Modify register.

Note:

register = DL, Q, S, Z, X, ST, P, 1, 2, 3, and 4 registers.

(cr) leaves value unchanged.
. terminates command.
ST.(2:7) may be changed.
DL ≤ 0 ≤ Q ≤ S ≤ Z

DL and Z expand and contract only in blocks of % 200 words, limited by MAXDATA, DL, and STACKSIZE parameters in :PREP, :PREPRUN, or :RUN commands.

If *register* omitted, all registers between DL and Z are modified.

Privileged Mode

All ST bits can be changed.

$ register := expression — Modify single register value.

Notes:
register may be ST, X, DL, Q, S, Z, P, or 1, 2, 3, 4 register.

expression signifies new value

Calculation Display Command

= expr [,mode] — Calculate and display *expr* value.

Notes:
expr=

Operators	Operands
*, /, +, –	[%] octal no.
Use ( ) to override hierarchy.	# decimal no.
	"[char] [char]"

mode= O (octal)
A (ASCII)
I (Decimal Integer)
H (hexadecimal)
C (Code)

Example of expr:

=#4+(55*#12)–"A",I

Trace Command

T — Trace stack markers.

Note:
Displays Q-displacement, LCST, P (relative) for markers not including initial stack marker. See Stack, page 113.

Privileged Mode

T also displays absolute CST.

Note:
Listing format for each marker is

$$Q\text{–}dq, LCST = \begin{bmatrix} P \\ G \\ S \end{bmatrix} lsn, P = pc, CST = asn$$

where:

dq = Displacement from current Q

P = Account Public SL

G = Group SL

S = System SL

lsn = Logical CST no.

pc = P (relative) address.

asn = Absolute (actual) CST (privileged mode only).

Segment Freeze Commands

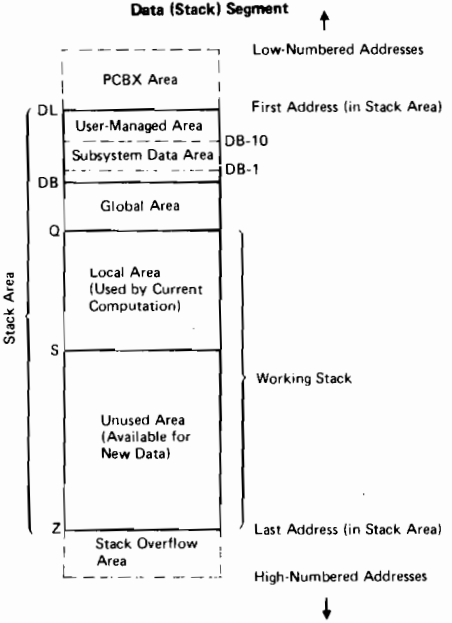
Privileged Mode

F {CO / DA} segmt — Freezes code or data segment in memory

U {CO / DA} segmt — Unfreezes frozen code or data segment.

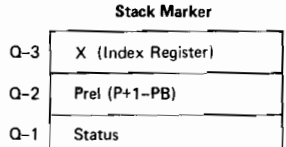
Note:
CO = Code segment
DA = Data segment.

SEGMENT AND REGISTER CONTENTS



DL - Data Limit
DB - Data Base (pointer to Global Area)
Q = Stack Marker
S = TOS
Z - Last Stack Address

STACK MARKER FORMAT



Q-3 = Current contents of X-register

Q-2 = Return address of code segment (P+1, PB rel.)

Q-1 = Current contents of status retgister.

Q-0 = Δ Q: the number of words between new and previous Q.

Note: *Parm* parameter of :RUN command found in Q-4 of outer block or main program.

PMAP FORMAT

```
PROGRAM FILE file.group.acct
Segname   lsn

   NAME     STT    CODE   ENTRY    SEG
   ep/pname en     begloc eploc    lsn-ep

   SEGMENT LENGTH  seglng

PRIMARY DB pdb INITIAL STACK is CAPABILITY cap
SECONDARY DB sdb INITIAL DL idl TOTAL CODE tc
TOTAL DB tdb MAXIMUM DATA mdl TOTAL REC  tr
ELAPSED TIME: et   PROCESSOR TIME: cpu
```

PMAP FORMAT

file.group.acct = Program file name.

segname = Segment name.

lsn = Logical segment no.

ep/pname = Name of program unit entry-point procedure.

en = Assigned entry no. in Segment Transfer Table.

begloc = Beginning location of procedure code in segment.

eploc = Location of entry-point in segment.

lsn-ep = Logical segment number of segment containing this external procedure.

seglng = Segment length (words)

pdb = Primary DB area size.

sdb = Secondary DB area size.

tdb = Total DB area size.

et = Preparation time elapsed (minutes).

is = Initial stack size.

idl = Initial DL size

mdl = Maximum area available for data (Z-DL).

cap = Program file capability.

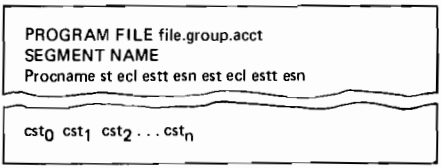
tc = Total code in file.

tr = Total records in file.

cpu = CPU time used for preparation.

Note: All numbers in octal.

LMAP Format



file.group.acct = Program file name.

Procname = External procedure name.

st = Type of segment referencing external procedure:

PROG = Program

GSL = Group Segment Library

PSL = Public Segment Library

ecl = External parameter checking level.

estt = External segment transfer table (STT) no.

esn = External logical segment no.

est = Entry point segment type:

GSL: Group Segment Library

PSL: Public Segment Library

SSL: System Segment Library

ecl = Entry-point parameter checking level.

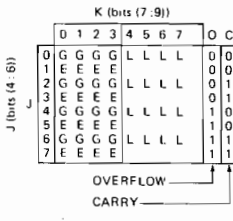
estt = Entry-point segment transfer table (SST) no.

esn = Entry-point logical segment no.

cst_0 . . . cst_n = List of code segment table nos. to which program file segments were assigned, ordered by logical segment no. (0, 1, 2, . . . n, reading left-to-right.)

Note: All numbers in octal.

CONDITION CODES



CCA

- L opr < 0
- E opr = 0
- G opr > 0

CCB

- L spec ASCII
- E Alphabetic
- G Numeric

CCC

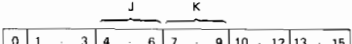
- L opr 1 < opr 2
- E opr 1 = opr 2
- G opr 1 > opr 2

Note: Position of J and K octal characters shown below.

CCD

- L Non-responding device controller.
- E Responding device controller, or device ready.
- G Device not ready (busy).

STATUS REGISTER



Numeric
G OP1>OP2
OP>0

Alpha
E OP1=OP2
OP=0

Special
L OP1>OP2
OP>0

6-7 CC field

0	> G
1	< L
2	= E
3	Undef

File System

Input Set

File Designator	Meaning
$STDIN	Session/job input device
$STDINX	Session/job input device with commands allowed.
$OLDPASS	Last $NEWPASS file closed.
$NULL	Constantly-empty file that returns EOF indication when read.
*formaldesignator	Back reference to previously defined file.
filereference	File name, indicates old file.

Output Set

File Designator	Meaning
$STDLIST	Session/job listing device
$OLDPASS	Last file passed.
$NEWPASS	New temporary file to be passed.
$NULL	Constantly-empty file that returns a successful indication whenever data is written to it.
*formaldesignator	Back reference to previously defined file.
filereference	File name, indicates a temporary new file sending on disc.

File Codes

Mnemonic	Integer	Meaning
USL	1024	A USL file.
BASD	1025	A BASIC data file.
BASP	1026	A BASIC program file.
BASFP	1027	A BASIC fast program file.
RL	1028	A relocatable library (RL) file.
PROG	1029	A program file.
	1030	A STAR file.
SL	1031	A segmented library (SL) file.
XLSAV	1040	A Cross Loader ASCII file (SAVE).
XLBIN	1041	A Cross Loader relocated binary file.
XLDSP	1042	A Cross Loader ASCII file (DISPLAY).
EDITQ	1050	An EDIT KEEPQ file (non-COBOL).
EDTCQ	1051	An EDIT KEEPQ file (COBOL).
EDTCT	1052	An EDIT TEXT file (COBOL).
RJEPN	1060	An RJE punch file.
	1069	An RSAM file.
QPROC	1070	A QUERY procedure file.
	1071 1072	QUERY work files.
KSAMK	1080	A KSAM key file.
LOG	1090	User Logging file

Default: 0.

Carriage-Control Directives

OCTAL CODE	ASCII SYMBOL	CARRIAGE ACTION
%40	" "	Single space (with or without automatic page eject).
%53	"+"	No space, return (next printing at column 1). Not valid on 2607 (results in single space without automatic page eject).
%53	"-"	Triple space (without automatic page eject).
%60	"0"	Double space (without automatic page eject).
%61	"1"	Page eject (form feed). Selects VFC Channel 1.
%2nn (nn is any octal number from 0 through 77)		Space nn lines (no automatic page eject). %200 not valid for 2607 (results in single space without automatic page eject).
%300-%307		Select VFC Channel 1-8 (2607)
%300-%313		Select VFC Channel 1-12 (2613, 2617, 2618, 2619)
%300-%317		Select VFC Channel 1-16 (2608)
		NOTE: Channel assignments shown below are the HP standard defaults.
%300		Skip to top of form (page eject).
%301		Skip to bottom of form.
%302		Single spacing with automatic page eject.
%303		Skip to next odd line with automatic page eject.
%304		Skip to next third line with automatic page eject.
%305		Skip to next 1/2 page.
%306		Skip to next 1/4 page.
%307		Skip to next 1/6 page.
%310		Skip to bottom of form.
%311		User option (2613/17/18/19), skip to one line before bottom of form (2608)
%312		User option (2613/17/18/19), skip to one line before top of form (2608)
%313		User option (2613/17/18/19), skip to top of form (2608)
%314		Skip to next seventh line with automatic page eject.

OCTAL CODE	ASCII SYMBOL	CARRIAGE ACTION
%315		Skip to next sixth line with automatic page eject.
%316		Skip to next fifth line with automatic page eject.
%317		Skip to next fourth line with automatic page eject.
%320		No space, no return (next printing physically follows this).
%2-%37		
%41-%52		
%54		
%56-%57		
%62-%77		Same as %40
%104-%177		
%310-%317 (2607)		
%314-%317 (2613/17/18/19)		
%321-%377		
%400 or %100		Sets post-space movement option; this first prints, then spaces. If previous option was pre-space movement, the driver outputs a line with a skip to VFC Channel 3 to clear the buffer.
%401 or %101		Sets pre-space movement option; this first spaces, then prints.
%402 or %102		Sets signal-space option, with automatic page eject (60 lines per page).
%403 or %103		Sets single-space option, without automatic page eject (66 lines per page).
%1001		Enables CONTINUOUS WRITE (privileged Mode Capability only).
%2001		Disables CONTINGUOUS WRITE (Privileged Mode Capability only).
		NOTE: All page ejects (codes %61, %300, and (for 2608) %313) are suppressed if the current request has a transfer count of 0 and the previous request ending with a page eject.

Carriage Control Effect Summary

FOPEN or :FILE	FWRITE *Control* Parameter		
	= 0	= 1	= Greater than 1
Carriage-Control Foption Specified or CCTL	Byte 1 — 133 — | 0 | record = 132 | — DATA — Data output contains 132 characters; the prefix byte is added and contains 0.	— 132 — | record = 132 | — DATA — Data output contains 132 characters; the carriage-control character in the first byte is not printed if output is to a list device.	Byte 1 — 133 — | con-trol | record = 132 | — DATA — Data output contains 132 characters; the prefix character added is the carriage-control character specified by the FWRITE control parameter.
Carriage Control Foption *not* Specified or NOCCTL	— 132 — | record = 132 | — DATA — Data output contains 132 characters.	— 132 — | record = 132 | — DATA — Data output contains 132 characters.	— 132 — | record = 132 | — DATA — Data output contains 132 characters.
	EFFECT ON DATA OUTPUT		

File System

File Access/Security

Mode:		User:	
R	Read	ANY	Any user
L	Lock	AL	Account Librarian
A	Append	GL	Group Librarian
W	Write	CR	Creating User
X	Execute	GU	Group User
S	Save (in group)	AC	Account Member

ACCOUNT, GROUP, AND FILE DEFAULT SECURITY

	Access Permitted
SYS Account	(R,X:ANY;W,A,L:AC)
Accounts other than SYS	(R,X,W,A,L:AC)
PUB Groups in any account	(R,X:ANY;A,W,L,S:AL, GU)
Groups other than PUB	(R,X,S,W,A,L:GU)
Files	(R,X,W,A,L:ANY)

NET DEFAULT ACCESS

Filereference	File	Access Permitted	Save Access to Group
filename. PUB.SYS	Any file in Public Group of System Account.	(R,X:ANY; W:AL, GU)	AL,GU
filename. PUB.account name	Any file in Public Group of any account.	(R,X:AC;W:AL, GU)	AL,GU
filename. groupname. account name	Any file in any group in any account	(R,W,X:GU)	GU

RUN TIME ERRORS

MSGNO	Message
0	END OF FILE
1	ILLEGAL DB REGISTER
2	ILLEGAL CAPABILITY
3	OMITTED PARAMETER
4	INCORRECT S REGISTER
5	PARAMETER ADDRESS VIOLATION
6	PARAMETER END ADDRESS VIOLATION
7	ILLEGAL PARAMETER
8	PARAMETER VALUE INVALID
9	INCORRECT Q REGISTER

FILE SYSTEM ERRORS

0	END OF FILE (FSERR 0)
1	ILLEGAL DB REGISTER SETTING (FSERR 1)
2	ILLEGAL CAPABILITY (FSERR 2)
3	REQUIRED PARAMETER IS MISSING (FSERR 3)
8	ILLEGAL PARAMETER VALUE (FSERR8)
9	INVALID FILE TYPE SPECIFIED IN FOPTIONS (FSERR 9)
10	INVALID RECORD SIZE SPECIFICATION
11	INVALID BLOCKSIZE
16	MORE THAN 255 OPENS OF A FILE (FSERR 16)
17	MAGNETIC TAPE RUNAWAY (FSERR 17)
18	DEVICE POWERED UP (FSERR 18)
19	FORMS CONTROL WAS RESET (FSERR 19)
20	INVALID OPERATION (FSERR 20)
21	DATA PARITY ERROR (FSERR 21)
22	SOFTWARE TIME-OUT (FSERR 22)
23	END OF TAPE (FSERR 23)
24	UNIT NOT READY (FSERR 24)
25	NO WRITE-RING ON TAPE (FSERR 25)
26	TRANSMISSION ERROR (FSERR 26)
27	I/O TIME-OUT (FSERR 27)
28	TIMING ERROR OR DATA OVERRUN (FSERR 28)
29	SIO FAILURE (FSERR 29)
30	UNIT FAILURE (FSERR 30)
31	END OF LINE (FSERR 31)
32	SOFTWARE ABORT (FSERR 32)
33	DATA LOST (FSERR 33)
34	UNIT NOT ON-LINE (FSERR 34)
35	DATA-SET NOT READY (FSERR 35)
36	INVALID DISC ADDRESS (FSERR 36)
37	INVALID MEMORY ADDRESS (FSERR 37)
38	TAPE PARITY ERROR (FSERR 38)
39	RECOVERED TAPE ERROR (FSERR 39)

40	OPERATION INCONSISTENT WITH ACCESS TYPE (FSERR 40)
41	OPERATION INCONSISTENT WITH RECORD TYPE (FSERR 41)
42	OPERATION INCONSISTENT WITH DEVICE TYPE (FSERR 42)
43	WRITE EXCEEDS RECORD SIZE (FSERR 43)
44	UPDATE AT RECORD ZERO (FSERR 44)
45	PRIVILEGED FILE VIOLATION (FSERR 45)
46	OUT OF DISC SPACE (FSERR 46)
47	I/O ERROR ON FILE LABEL (FSERR 47)
48	INVALID OPERATION DUE TO MULTIPLE FILE ACCESS (FSERR 48)
49	UNIMPLEMENTED FUNCTION (FSERR 49)
50	NONEXISTENT ACCOUNT (FSERR 50)
51	NONEXISTENT GROUP (FSERR 51)
52	NONEXISTENT PERMANENT FILE (FSERR 52)
53	NONEXISTENT TEMPORARY FILE (FSERR 53)
54	INVALID FILE REFERENCE (FSERR 54)
55	DEVICE UNAVAILABLE (FSERR 55)
56	INVALID DEVICE SPECIFICATION (FSERR 56)
57	OUT OF VIRTUAL MEMORY (FSERR 57)
58	NO PASSED FILE (FSERR 58)
59	STANDARD LABEL VIOLATION (FSERR 59)
60	GLOBAL RIN UNAVAILABLE (FSERR 60)
61	OUT OF GROUP DISC SPACE (FSERR 61)
62	OUT OF ACCOUNT DISC SPACE (FSERR 62)
63	USER LACKS NON-SHARABLE DEVICE CAPABILITY (FSERR 63)
64	USER LACKS MULTI-RIN CAPABILITY (FSERR 64)
65	PUNCH HOPPER EMPTY (FSERR 65)

66	PLOTTER LIMIT SWITCH REACHED (FSERR 66)
67	PAPER TAPE ERROR (FSERR 47)
68	INSUFFICIENT SYSTEM RESOURCES (FSERR 68)
69	I/O ERROR (FSERR 69)
70	I/O ERROR WHILE PRINTING HEADER/ TRAILER (FSERR 70)
71	TOO MANY FILES OPEN (FSERR 71)
72	INVALID FILE NUMBER (FSERR 72)
73	BOUNDS VIOLATION (FSERR 73)
77	NO-WAIT I/O PENDING (FSERR 77)
78	NO NO-WAIT I/O PENDING FOR ANY FILE (FSERR 78)
79	NO NO-WAIT I/O PENDING FOR SPECIAL FILE (FSERR 79)
80	SPOOFLE SIZE EXCEEDS CONFIGURATION (FSERR 80)
81	NO "SPOOL" CLASS IN SYSTEM (FSERR 81)
82	INSUFFICIENT SPACE FOR SPOOFLE (FSERR 82)
83	I/O ERROR ON SPOOFLE (FSERR 83)
84	DEVICE UNAVAILABLE FOR SPOOFLE (FSERR 84)
85	OPERATION INCONSISTENT WITH SPOOLING (FSERR 85)
86	SPOOLING INTERNAL ERROR (FSERR 86)
87	BAD SPOOFLE BLOCK (FSERR 87)
88	SPOOLING ERROR (FSERR 88)
89	POWER FAILURE (FSERR 89)
90	EXCLUSIVE VIOLATION: FILE BEING ACCESSED (FSERR 90)
91	EXCLUSIVE VIOLATION: FILE ACCESSED EXCLUSIVELY (FSERR 91)
92	LOCKWORD VIOLATION (FSERR 92)
93	SECURITY VIOLATION (FSERR 93)
94	USER IS NOT CREATOR (FSERR 94)
95	READ COMPLETED DUE TO BREAK (FSERR 95)
96	DISC I/O ERROR (FSERR 96)
97	NO CONTROL Y PIN (FSERR 97)
98	READ TIME OVERFLOW (FSERR 98)
99	EOT AND BACKSPACE TAPE (FSERR 99)

100	DUPLICATE PERMANENT FILE NAME (FSERR 100)
101	DUPLICATE TEMPORARY FILE NAME (FSERR 101)
102	I/O ERROR ON DIRECTORY (FSERR102)
103	PERMANENT FILE DIRECTORY OVERFLOW (FSERR 103)
104	TEMPORARY FILE DIRECTORY OVERFLOW (FSERR 104)
105	BAD VARIABLE BLOCK STRUCURE (FSERR 105)
106	EXTENT SIZE EXCEEDS MAXIMUM (FSERR 106)
107	INSUFFICIENT SPACE FOR USER LABELS (FSERR 107)
108	INVALID FILE LABEL (FSERR 108)
109	INVALID CARRIAGE CONTROL (FSERR 109)
110	ATTEMPT TO SAVE PERMANENT FILE AS TEMPORARY (FSERR 110)
111	USER LACKS SAVE FILES (SF) CAPABILITY (FSERR 111)
112	USER LACKS PRIVATE VOLUMES (UV) CAPABILITY (FSERR 112)
113	VOLUME SET NOT MOUNTED – MOUNT PROBLEM (FSERR 113)
114	VOLUME SET NOT DISMOUNTED – DISMOUNT PROBLEM (FSERR 114)
115	ATTEMPTED RENAME ACROSS VOLUME SETS – REJECTED (FSERR 115)
116	INVALID TAPE LABEL FOPEN PARAMETERS (FSERR 116)
117	ATTEMPT TO WRITE ON AN UNEXPIRED TAPE FILE (FSERR 117)
118	INVALID HEADER OR TRAILER TAPE LABEL (FSERR 118)
119	I/O ERROR POSITIONING TAPE FOR TAPE LABELS (FSERR 119)
120	ATTEMPT TO WRITE IBM STANDARD TAPE LABEL (FSERR 120)
121	TAPE LABEL LOCKWORD VIOLATION (FSERR 121)
122	TAPE LABEL TABLE OVERFLOW (FSERR 122)

123	END OF TAPE VOLUME SET (FSERR 123)
124	ATTEMPT TO APPEND LABELED TAPE (FSERR 124)
126	CHARACTER SET NUMBER MUST BE BETWEEN 0 AND 31 (FSERR 126)
127	FORM NUMBER MUST BE BETWEEN 0 AND 31 (FSERR 127)
128	LOGICAL PAGE NUMBER MUST BE BETWEEN 0 AND 31 (FSERR 128)
129	VERTICAL FORMAT NUMBER MUST BE BETWEEN 0 AND 31 (FSERR 129)
130	NUMBER OF COPIES MUST BE BETWEEN 1 AND 32767 (FSERR 130)
131	NUMBER OF OVERLAYS MUST BE BETWEEN 1 AND 8 (FSERR 131)
132	PAGE LENGTH PARM MUST BE BETWEEN 12 (=3") and 68 (=17") (FSERR 132)
137	DEFECTIVE TRACK ON FOREIGN DISC (FSERR 137)
138	TRACK DOES NOT EXIST ON FOREIGN DISC (FSERR 138)
139	DELETED RECORD ON IBM DISKETTE (FSERR 139)
148	INACTIVE RIO RECORD (FSERR 148)
149	MISSING ITEM NUMBER OR RETURN-VARIABLE (FSERR 149)
150	INVALID ITEM NUMBER (FSERR 150)
151	UNDEFINED FILE TYPE (FSERR 151)
152	UNRECOGNIZED KEYWORD IN FOPEN DEVICE PARAMETER (FSERR 152)
153	EXPECTED ";" OR CARRIAGE RETURN IN DEVICE PARAMETER (FSERR 153)
154	ENVIRONMENT FILE OPEN ERROR (FSERR 154)
155	NOT ENVIRONMENT FILE. CHECK FILE CODE OR RECORD SIZE (FSERR 155)
156	ENVIRONMENT HEADER RECORD INCORRECT (FSERR 156)
157	UNCOMPILED ENVIRONMENT FILE (FSERR 157)
158	ERROR READING ENVIRONMENT FILE (FSERR 158)
159	ERROR CLOSING ENVIRONMENT FILE (FSERR 159)
160	ERROR DOING FDEVICECONTROL FROM ENVIRONMENT (FSERR 160)

161 TOO MANY PARAMETERS IN DEVICE STRING – OVERFLOW (FSERR 161)
162 EXPECTED "=" AFTER KEYWORD IN DEVICE PARAMETER (FSERR 162)
163 "ENV" BACK REFERENCE IN FILE EQUATION INCORRECT (FSERR 163)
164 DEVICE PARAMETER TOO LARGE OR MISSING CARRIAGE RETURN (FSERR 164)
165 INVALID DENSITY SPECIFICATION (FSERR 165)
166 FFILEINFO FAILED IN ACCESSING REMOTE SPOOL FILE (FSERR 166)
167 FILE LABEL ERROR IN SPOOL FILE. CANNOT INSERT ENVIRONMENT & FILE NAME (FSERR 167)
171 DUPLICATE KEY VALUE (FSERR 171)
172 NO SUCH KEY (FSERR 172)
173 TCOUNT PARAMETER LARGER THAN RECORD SIZE (FSERR 173)
174 CANNOT GET EXTRA DATA SEGMENT (FSERR 174)
175 KSAN INTERNAL ERROR (FSERR 175)
176 ILLEGAL EXTRA DATA SEGMENT LENGTH (FSERR 176)
177 TOO MANY EXTRA DATA SEGMENTS FOR THIS PROCESS (FSERR 177)
178 EXTRA DATA SEGMENT TOO SMALL (FSERR 178)
179 THE FILE MUST BE LOCKED BEFORE ISSUING THIS INTRINSIC (FSERR 179)
181 INVALID KEY STARTING POSITION (FSERR 181)
182 FILE IS EMPTY (FSERR 182)
183 RECORD DOES NOT CONTAIN ALL KEYS (FSERR 183)
184 INVALID RECORD NUMBER (FFINDN INTRINSIC) (FSERR 184)
185 SEQUENCE ERROR IN PRIMARY KEY (FSERR 185)
186 INVALID KEY LENGTH – NUMERIC DISPLAY OR PACKED DECIMAL (FSERR 186)
187 INVALID KEY SPECIFICATION (FSERR 187)
188 INVALID DEVICE SPECIFICATION (FSERR 188)

189	INVALID RECORD FORMAT (FSERR 189)
190	INVALID KEY BLOCKING FACTOR VALUE (FSERR 190)
191	RECORD DOES NOT CONTAIN SEARCH KEY SPECIFIED FOR DELETION (FSERR 191)
192	SYSTEM FAILURE OCCURRED WHILE THE KSAM FILE WAS OPENED (FSERR 192)
201	INVALID ID SEQUENCE (FSERR 201)
202	INVALID TELEPHONE NUMBER (FSERR 202)
203	NO TELEPHONE LIST SPECIFIED (FSERR 203)
204	UNABLE TO ALLOCATE AN EXTRA DATA SEGMENT FOR DS/3000 (DSERR 204)
205	UNABLE TO EXPAND THE DS/3000 EXTRA DATA SEGMENT (DSERR 205)
206	SLAVE PTOP FUNCTION ISSUED FROM A MASTER PROGRAM. (DSERR 206)
207	SLAVE PTOP FUNCTION OUT OF SEQUENCE. (DSERR 207)
208	MASTER PTOP FUNCTION ISSUED BY A SLAVE PROGRAM. (DSERR 208)
209	SLAVE PROGRAM DOES NOT EXIST OR IS NOT PROGRAM FILE. (DSERR 209)
210	WARNING – INVALID MAXDATA OR DLSIZE FOR A SLAVE PROGRAM. SYSTEM DEFAULTS ARE IN EFFECT. (DSWARN 210)
211	SLAVE ISSUED A REJECT TO A MASTER PTOP OPERATION. (DSWARN 211)
212	FILE NUMBER RETURNED FROM IOWAIT IS NOT A DS LINE NUMBER (DSWARN 212)
213	EXCLUSIVE USE OF A DS LINE REQUIRES BOTH ND AND CS CAPABILITY (DSERR 213)
214	THE REQUESTED DS LINE HAS NOT BEEN OPEN WITH A USER :DSLINE COMMAND (DSERR 214)
215	DSLINE CANNOT BE ISSUED BACK TO THE MASTER COMPUTER. (DSERR 215)

216	MESSAGE REJECTED BY THE REMOTE COMPUTER (DSERR 216)
217	INSUFFICIENT AMOUNT OF USER STACK AVAILABLE (DSERR 217)
218	INVALID PTOP FUNCTION REQUESTED. (DSERR 218)
219	MULTIPLE POPEN. ONLY ONE MASTER PTOP OPERATION CAN BE ACTIVE ON A DS LINE. (DSERR 219)
220	PROGRAM EXECUTING GET WAS NOT CREATED BY POPEN. (DSERR 220)
221	INVALID DS MESSAGE FORMAT (INTERNAL DS ERROR) (DSERR 221)
222	MASTER PTOP FUNCTION ISSUED PRIOR TO A POPEN. (DSERR 222)
223	REQUEST TO SEND MORE DATA THAN SPECIFIED IN POPEN. (DSERR 223)
224	FILE EQUATIONS FOR A REMOTE FILE CONSTITUTE A LOOP. (DSERR 224)
225	CANNOT ISSUE POPEN TO A SLAVE SESSION IN BREAK MODE. (DSERR 225)
226	SLAVE PROGRAM HAS TERMINATED BEFORE EXECUTING "GET". (DSERR 226)
227	REMOTE HELLO MUST BE DONE TO INITIATE REMOTE SESSION. (DSERR 227)
236	COMMUNICATIONS HARDWARE HAS DETECTED AN ERROR. (DSERR 236)
237	CANNOT CURRENTLY GAIN ACCESS TO THE TRACE FILE. (DSERR 237)
238	COMMUNICATIONS INTERFACE ERROR. INTERNAL FAILURE. (DSERR 238)
239	COMMUNICATIONS INTERFACE ERROR. TRACE MALFUNCTION. (DSERR 239)
240	THE LOCAL COMMUNICATION LINE HAS NOT BEEN OPENED BY THE OPERATOR (DSERR 240)
241	THE DS LINE IS IN USE EXCLUSIVELY OR BY ANOTHER SUBSYSTEM (DSERR 241)
242	INTERNAL DS SOFTWARE MALFUNCTION (DSERR 242)

243	THE REMOTE COMPUTER IS NOT RESPONDING (DSERR 243)
244	COMMUNICATIONS INTERFACE ERROR. THE REMOTE COMPUTER RESET THE LINE (DSERR 244)
245	COMMUNICATIONS INTERFACE ERROR. RECEIVE TIMEOUT (DSERR 245)
246	COMMUNICATIONS INTERFACE ERROR. REMOTE HAS DISCONNECTED (DSERR 246)
247	COMMUNICATIONS INTERFACE ERROR. LOCAL TIME OUT (DSERR 247)
248	COMMUNICATIONS INTERFACE ERROR. CONNECT TIME OUT (DSERR 248)
249	COMMUNICATIONS INTERFACE ERROR. REMOTE REJECTED CONNECTION (DSERR 249)
250	COMMUNICATIONS INTERFACE ERROR. CARRIER LOST (DSERR 250)
251	COMMUNICATIONS INTERFACE ERROR. THE LOCAL DATA SET FOR THE DS LINE WENT NOT READY (DSERR 251).
252	COMMUNICATIONS INTERFACE ERROR. HARDWARE FAILURE (DSERR 252)
253	COMMUNICATIONS INTERFACE ERROR. NEGATIVE RESPONSE TO THE DIAL REQUEST BY THE OPERATOR (DSERR 253)
254	COMMUNICATIONS INTERFACE ERROR. INVALID I/O CONFIGURATION (DSERR 254)
255	COMMUNICATIONS INTERFACE ERROR. UNANTICIPATED ERROR CONDITION (DSERR 255)
300	NUMBER OF OPENS FOR FILE EXCEEDS 255. (FSERR 300)
301	FREE SPACE TABLE FOR LDEV ! IS FULL, RUN VINIT -COND

Section XI

ASCII, Instruction Set

ASCII CHARACTER SET/COLLATING SEQUENCE

BYTE POSITION			
CHAR	Left	Right	Dec.
NUL	000000	000000	0
SOH	000400	000001	1
STX	001000	000002	2
ETX	001400	000003	3
EOT	002000	000004	4
ENQ	002400	000005	5
ACK	003000	000006	6
BEL	003400	000007	7
BS	004000	000010	8
HT	004400	000011	9
LF	005000	000012	10
VT	005400	000013	11
FF	006000	000014	12
CR	006400	000015	13
SO	007000	000016	14
SI	007400	000017	15
DLE	010000	000020	16
DC1	010400	000021	17
DC2	011000	000022	18
DC3	011400	000023	19
DC4	012000	000024	20
NAK	012400	000025	21
SYN	013000	000026	22

BYTE POSITION			
CHAR	Left	Right	Dec.
!	026400	000055	45
"	027000	000056	46
/	027400	000057	47
0	030000	000060	48
1	030400	000061	49
2	031000	000062	50
3	031400	000063	51
4	032000	000064	52
5	032400	000065	53
6	033000	000066	54
7	033400	000067	55
8	034000	000070	56
9	034400	000071	57
:	035000	000072	58
;	035400	000073	59
<	036000	000074	60
=	036400	000075	61
>	037000	000076	62
?	037400	000077	63
@	040000	000100	64

BYTE POSITION			
CHAR	Left	Right	Dec.
[	055400	000133	91
\	056000	000134	92
]	056400	000135	93
^	057000	000136	94
_	057400	000137	95
`	060000	000140	96
a	060400	000141	97
b	061000	000142	98
c	061400	000143	99
d	062000	000144	100
e	062400	000145	101
f	063000	000146	102
g	063400	000147	103
h	064000	000150	104
i	064400	000151	105
j	065000	000152	106
k	065400	000153	107
l	066000	000154	108
m	066400	000155	109
n	067000	000156	110
o	067400	000157	111

ASCII Character Set

ASCII CHARACTER SET/COLLATING SEQUENCE

ETB	013400	000027	23
CAN	014000	000030	24
EM	014400	000031	25
SUB	015000	000032	26
ESC	015400	000033	27
FS	016000	000034	28
GS	016400	000035	29
RS	017000	000036	30
US	017400	000037	31
SPACE	020000	000040	32
!	020400	000041	33
"	021000	000042	34
#	021400	000043	35
$	022000	000044	36
%	022400	000045	37
&	023000	000046	38
'	023400	000047	39
(	024000	000050	40
)	024400	000051	41
*	025000	000052	42
+	025400	000053	43
,	026000	000054	44

A	040400	000101	65
B	041000	000102	66
C	041400	000103	67
D	042000	000104	68
E	042400	000105	69
F	043000	000106	70
G	043400	000107	71
H	044000	000110	72
I	044400	000111	73
J	045000	000112	74
K	045400	000113	75
L	046000	000114	76
M	046400	000115	77
N	047000	000116	78
O	047400	000117	79
P	050000	000120	80
Q	050400	000121	81
R	051000	000122	82
S	051400	000123	83
T	052000	000124	84
U	052400	000125	85
V	053000	000126	86
W	053400	000127	87
X	054000	000130	88
Y	054400	000131	89
Z	055000	000132	90

p	070000	000160	112
q	070400	000161	113
r	071000	000162	114
s	071400	000163	115
t	072000	000164	116
u	072400	000165	117
v	073000	000166	118
w	073400	000167	119
x	074000	000170	120
y	074400	000171	121
z	075000	000172	122
{	075400	000173	123
|	076000	000174	124
}	076400	000175	125
~	077000	000176	126
DEL	077400	000177	127

Index of OP Code Groups

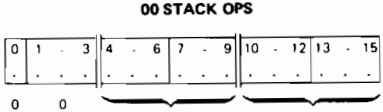
ADAX	0	CPRB	1	DZRO	0
ADBX	0	CSL	1	EADD	2
ADD	0	CSR	1	ECMP	2
ADDD	2	CVAD	2	EDIV	2
ADDI	2	CVBD	2	EMPY	2
ADDM	4	CVDA	2	ENEG	2
ADDS	3	CVDB	2	ESUB	2
ADXA	0	DABZ	1	EXF	2
ADXB	0	DADD	0	EXIT	3
ADXI	2	DASL	1	FADD	0
AND	0	DASR	1	FCMP	0
ANDI	3	DCMP	0	FDIV	0
ASL	1	DCSL	1	FIXR	0
ASR	1	DCSR	1	FIXT	0
		DDEL	0	FLT	0
BCY	1	DDIV	2	FMPY	0
BE	4	DDUP	0	FNEG	0
BG	4	DECA	0	FSUB	0
BGE	4	DECB	0	HALT	3
BL	4	DECM	4	IABZ	1
BLE	4	DECX	0	INCA	0
BNE	4	DEL	0	INCB	0
BNCY	1	DELB	0	INCM	4
BNOV	1	DFLT	0	INCX	0
BOV	1	DISP	3	IXBZ	1
BR	4	DIV	0	IXIT	2
BRE	1	DIVI	2	LADD	0
BRO	1	DIVL	0	LCMP	0
BTST	0	DLSL	1	LDB	4
CAB	0	DLSR	1	LDD	4
CIO	3	DMPY	2	LDEA	2
CMD	3	DNEG	0	LDI	2
CMP	0	DPF	2	LDIV	0
CMPB	8 0	DSUB	0	LDNI	2
CMPD	2	DTST	0	LDPN	3
CMPI	2	DUP	0	LDPP	3
CMPM	4	DXBZ	1	LDX	4
CMPN	2	DXCH	0	LDXA	0

Index of OP Code Groups (cont.)

LDXB	0	NSLD	2	STAX	0
LDXI	2	OR	0	STB	4
LDXN	2	ORI	3	STBX	0
LLBL	3	PAUS	3	STD	4
LLSH	2	PCAL	3	STOR	4
LMPY	0	PCN	2	SUB	0
LOAD	4	PLDA	2	SUBD	2
LOCK	2	PSDB	3	SUBI	2
LRA	4	PSEB	3	SUBM	4
LSEA	2	PSHR	2	SUBS	3
LSL	1	PSTA	2	SXIT	3
LSR	1	QASL	2	TASL	1
LST	3	QASR	2	TASR	1
LSUB	0	RCLK	3	TBA	4
MABS	2	RIO	3	TBC	1
MDS	2	RMSK	3	TBX	4
MFDS	2	RSW	2	TCBC	1
MOVE	2	SBXI	3	TEST	0
MPY	0	SCAL	3	TIO	3
MPYD	2	SCAN	1	TNSL	1
MPYI	2	SCLK	3	TRBC	1
MPYL	0	SCU	2	TSBC	1
MPYM	4	SCW	2	UNLK	2
MTBA	4	SDEA	2	WIO	3
MTBX	4	SED	3	XAX	0
MTDS	3	SETR	2	XBX	0
MVB	2	SIN	3	XCH	0
MVBL	2	DIO	3	XCHD	3
MVBW	2	SLD	2	XEQ	3
MVLB	2	SMSK	3	XOR	0
NEG	0	SRD	2	XORI	3
NOP	0	SSEA	2	ZERO	0
NOT	0	SST	3	ZROB	0
				ZROX	0

Instructions

00 STACK OPS



CC = condition, O = overflow, C = carry, CI = clears bit

CC	O	C			CC	O	C		
			00	NOP				40	DEL
			01	DELB				41	ZROB
			02	DDEL	A			42	LDXB
			03	ZROX	A			43	STAX
A	X	X	04	INCX	A			44	LDXA
A	X	X	05	DECX	A			45	DUP
			06	ZERO	A			46	DDUP
			07	DZRO	A			47	FLT
C			10	DCMP	C			50	FCMP
A	X	X	11	DADD	A	X		51	FADD
A	X	X	12	DSUB	A	X		52	FSUB
A	CI	X	13	MPYL	A	X		53	FMPY
A	X		14	DIVL	A	X		54	FDIV
A	X		15	DNEG	A			55	FNEG
A			16	DXCH	A			56	CAB
C			17	CMP	C			57	LCMP
A	X	X	20	ADD	A		X	60	LADD
A	X	X	21	SUB	A		X	61	LSUB
A	X		22	MPY	A		X	62	LMPY
A	X		23	DIV	A	X		63	LDIV
A	X	X	24	NEG	A			64	NOT
A			25	TEST	A			65	OR
A			26	STBX	A			66	XOR
A		X	27	DTST	A			67	AND
A			30	DFLT	A	X	X	70	FIXR
B			31	BTST	A	X	X	71	FIXT
A			32	XCH				72	Reserved
A	X	X	33	INCA	A	X	X	73	INCB
A	X	X	34	DECA	A	X	X	74	DECB
A			35	XAX				75	XBX
A	X	X	36	ADAX	A	X	X	76	ADBX
A	X	X	37	ADXA	A	X	X	77	ADXB

01 SHIFTS/BRANCHES

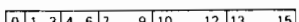
0	1	3	4	6	7	9	10	12	13	15
	0	1								

A			X	0	0	0	←SC→	ASL
A			X	0	0	1	←SC→	ASR
A			X	0	0	2	←SC→	LSL
A			X	0	0	3	←SC→	LSR
A			X	0	0	4	←SC→	CSL
A			X	0	0	5	←SC→	CSR
A			X	0	0	6	0 0	SCAN
A	X	X	I	0	0	7	+/- ← P branch →	IABZ
A			X	0	1	0	←SC→	TASL
A			X	0	1	1	←SC→	TASR
A	X	X	I	0	1	2	+/- ← P branch →	IXBZ
A	X	X	I	0	1	3	+/- ← P branch →	DXBZ
		CI	I	0	1	4	+/- ← P branch →	BCY
		CI	I	0	1	5	+/- ← P branch →	BNCY
A			X	0	1	6	0 0	TNSL
A				0	1	7	←SC→	QASL X
A							←SC→	QASR X
A			X	1	0	0	←SC→	DASL
A			X	1	0	1	←SC→	DASR
A			X	1	0	2	←SC→	DLSL
A			X	1	0	3	←SC→	DLSR
A			X	1	0	4	←SC→	DCSL
A			X	1	0	5	←SC→	DCSR
*			I	1	0	6	+/- ← P branch →	CPRB§
A	X	X	I	1	0	7	+ - ← P branch →	DABZ
	CI		I	1	1	0	+ - ← P branch →	BOV
	CI		I	1	1	1	+ - ← P branch →	BNOV
**			X	1	1	2	bit position	TBC
**			X	1	1	3	bit position	TRBC
**			X	1	1	4	bit position	TSBC
**			X	1	1	5	bit position	TCBC
			I	1	1	6	+ - ← P branch →	BRO
			I	1	1	7	+ - ← P branch →	BRE

C	O	C
C		

CI = clears bit
§ = uses Index Reg
** bit = 0 CCE, bit = 1 CCG or CCL
SC = shift count (0 - 63)
P branch signed magnitude (0 - 31)
bit position (0 - 63) [MOD 16]
CPRB X > (S) CCG X < (S-1) CCL
(S-1) <= X <= (S) CCE

02 MOVES/IMMEDIATES



				0	0	0 PB/DB 0		SDEC	MOVE
				0	0	1 PB/DB 0		SDEC	MVB
				0	1	0		SDEC	MVBL
				0	1	1		SDEC	MABS
B				0	1	2		SDEC	SCW
				0	1	3		SDEC	MTDS
				0	1	4		SDEC	MVLB
				0	1	5		SDEC	MDS
				0	1	6		SDEC	SCU
				0	1	7		SDEC	MFDS
B				0	2	0 N/A	U	SDEC	MVBW
C				0	2	1 PB/DB 0		SDEC	CMPB
A				0	3	0		0	RSW
A				0	3	0		1	LLSH§
A				0	3	2		0	PLDA§
				0	3	2		1	PSTA§
A				0	3	4		0	LSEA
				0	3	4		1	SSEA
A				0	3	4		2	LDEA
				0	3	4		3	SDEA
				0	3	6		0	IXIT
				0	3	6		1	**Reserved**
				0	3	6		2	PCN
				0	3	6		3	**Reserved**
A	X			0	4	1		0	EADD
A	X			0	4	1		1	ESUB
A	X			0	4	1		2	EMPY
A	X			0	4	1		3	EDIV
A				0	4	1		4	ENEG
C				0	4	1		5	ECMP
				0	5	7		0	DMUL
				0	5	7		1	DDIV
A		X		0	6	0		1	DMPY

02 MOVES/IMMEDIATES (cont.)

			0	1 - 3	4 - 6	7 - 9	10 - 12	13 - 15	
A	X				0	6	0	2	CVAD
A	X				0	6	0	3	CVDA
A	X				0	6	0	4	CVBD
A	X		0	2	0	6	0	5	CVDB
A	X	X			0	6	0	6	SLD
A	X	X			0	6	0	7	NSLD
A	X				0	6	1	0	SRD
A	X				0	6	1	1	ADDD
C	X				0	6	1	2	CMPD
A	X				0	6	1	3	SUBD
A	X				0	6	1	4	MPYD
A					1	0	← Imm Opr →		LDI
A					1	1	← Imm Opr →		LDXI
C					2	0	← Imm Opr →		CMPI
A	X	X			2	1	← Imm Opr →		ADDI
A	X	X			3	0	← Imm Opr →		SUBI
A	X				3	1	← Imm Opr →		MPYI
A					4	0	← Imm Opr →		DIVI
					4	1	SBK DB DL Z STA X Q S		PSHR *
A					5	0	← Imm Opr →		LDNI
A					5	1	← Imm Opr →		LDXN
C					6	0	← Imm Opr →		CMPN
A					6	1 J J	J J K	K K K	EXF
A					7	0 J J	J J K	K K K	DPF
					7	1 SBK DB	DL Z STA	X Q S	SETR
C	O	C							
C									

* = uses Index Register
SDEC pop stack (0 - 3)
Imm Opr Immediate operand (0 - 255)
JJJJ Beginning bit position (0 - 15)
KKKK field length (0 - 15)
* SCW CARRY → Terminating character
* SCU CARRY → Terminating character

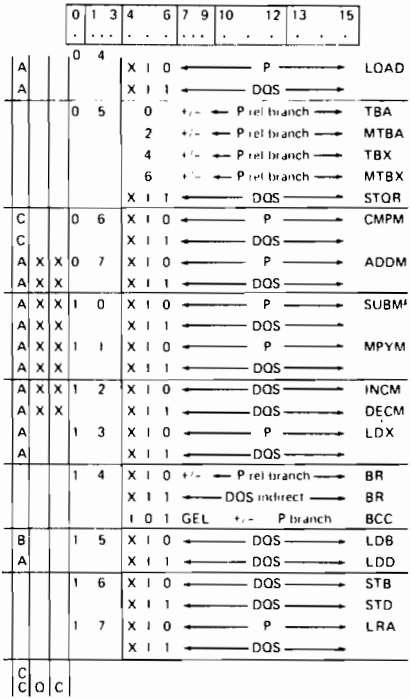
03 I/O LINKAGE CONTROL

0	1 - 3	4 - 6	7 - 9	10 - 12	13 - 15	
	0 3					
		0	0	0 0 K	K K K	LST
		0	0	0 1 **0**	**0 0 0**	PAUS
		0	0	1 0 0	0 0 D/E	SED
		0	0	6	0	XCHD
3		0	0	6	1	PSDB
		0	0	6	2	DISP
		0	0	6	3	PSEB
3		0	1	0	0	SMSK
		0	1	0	1	SCLK
		0	1	2	0	RMSK
		0	1	2	1	RCLK
*		0	1	1 0 K	K K K	XEQ
D		0	1	1 1 K	K K K	SIO
D		0	2	0 0 K	K K K	RIO
D		0	2	0 1 K	K K K	WIO
1		0	2	1 0 K	K K K	TIO
1		0	2	1 1 K	K K K	CIO
		0	3	0 0 K	K K K	CMD
		0	3	0 1 K	K K K	SST
1		0	3	1 0 K	K K K	SIN
		0	3	1 1 **K**	**K K K**	HALT
		0	1	← STT →		SCAL
		1	0	← STT →		PCAL
		1	1	← SDEC + (4) →		EXIT
		2	0	← SDEC + (1) →		SXIT
A		2	1	← Imm Opr →		ADXI
A		3	0	← Imm Opr →		SBXI
		3	1	← PL - Disp →		LLBL
		4	0	← P - Disp →		LDPP
		4	1	← P - Disp →		LDPN
		5	0	← Imm Opr →		ADDS
		5	1	← Imm Opr →		SUBS
		6	0	0	0	Reserved
A		6	1	← Imm Opr →		ORI
A		7	0	← Imm Opr →		XORI
A		7	1	← Imm Opr →		ANDI

C	O	C
C		

KKK Stack displacement (0 - 15)
STT entry position (0 - 255)
Imm Opr Immediate operand (0 - 255)
1. normally CCE, non-responding CCL
D norm CCE, non-respond CCL, not ready CCG
3. norm CCE, IF error CCL
* XEQ depends on instruction executed.

MEMORY REFERENCE



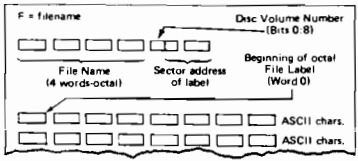
7 8 9 10 - 12 13 - 15

P	P+	0	0:377
	P-	1	0:377
DQS	DB+	0	0:377
	Q+	1 0	0:177
	Q-	1 1 0	0:77
	S-	1 1 1	0:77

GEL
1. Less than (BL)
2. Equal (BE)
3. Less than or equal (BLE)
4. Greater than (BG)
5. Not equal (BNE)
6. Greater or equal (BGE)

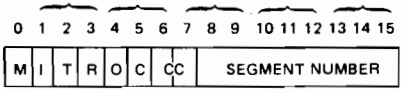
FILE LABELS

Format of LISTF-1 Listing



Words Dec	Words Octal		Contents
0-3	0-3		Local file name.
4-7	4-7		Group name.
8-11	10-13		Account name.
12-15	14-17		Identity of file creator.
16-19	20-23		File lockword.
20-21	24-25		File security matrix.
22	26	(Bits 0:15)	Not used.
		(Bit 15:1)	File secure bit: If 1, file secured. If 0, file released.
23	27		File creation date.
24	30		Last access date.
25	31		Last modification date.
26	32		File code.
27	33		File control block vector.
28	34	(Bit 0:1)	Store Bit. (If on, :STORE or :RESTORE, in progress.)
		(Bit 1:1)	Restore Bit. (If on, :RESTORE in progress.)
		(Bit 2:1)	Load Bit. (If on, program file is loaded.)
		(Bit 3:1)	Exclusive Bit. (If on, file is opened with exclusive access.)
		(Bits 4:4)	Device sub-type.
		(Bits 8:6)	Device type.
		(Bit 14:1)	File is open for write.
		(Bit 15:1)	File is open for read.
29	35	(Bits 0:8)	Number of user labels written.
		(Bits 8:8)	Number of user labels.
30-31	36-37		Maximum number of logical records.
32-33	40-41		Private Volume Information.
34	42		Checksum
35	43		Cold-load identity.
36	44		Foptions specifications.
37	45		Logical record size (in negative bytes).
38	46		Block size (in words).
39	47	(Bits 0:8)	Sector offset to data.
		(Bits 8:3)	Not used.
		(Bits 11:5)	Number of extents minus 1.
40	50		Logical size of last block.
41	51		Extent size.
42-43	52-53		Number of logical records in file.
See Appendix B in Intrinsic Manual			
44-107	54-153		Two-word addresses of up to 32 disc extents, beginning with address of first extent (words 44:45).
125-128	175-200		Device Class.

STATUS REGISTER



M Mode User Privileged
I Ext Interrupts Enabled
T User Traps Enabled
R Right Stack Op Pending
O Overflow
C Carry
CC Condition Code
CCL = 1
CCE = 2
CCG = 0

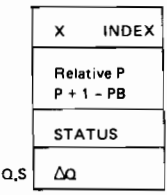
CONDITION CODES

CCA	CCL	Operand < 0
	CCE	Operand = 0
	CCG	Operand > 0
CCB	CCL	Special ASCII Char
	CCE	Alphabetic
	CCG	Numeric
CCC	CCL	Operand 1 < OPR 2
	CCE	Operand 1 = OPR 2
	CCG	Operand 1 > OPR 2
CCD	CCL	Non-responding device controller
	CCE	Responding device controller, or device ready
	CCG	Device not ready (busy)

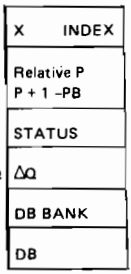
CC FIELD

6-7	Condition
0	>G
1	<L
2	= E
3	Reserved

STACK MARKER (normal)



SPECIAL STACK MARKER (resulting from ICS interrupt)



CODE SEGMENT TABLE

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	M	R	T	LENGTH											
RESERVED															
													PB BANK		
ADDRESS															

A Absence bit (=1 if segment is absent)
M Mode bit (=1 if privileged mode)
R Reference bit (for statistical use by operating system, set to 1 when accessed)
T Trace bit (=1 to call Trace routine)
LENGTH This value times 4 (max = 16,380)
ADDRESS Absolute memory address (for PB) or low order 16 bits of absolute disc address if absent.
PB BANK Bank Number if present of High Order Disc Address if absent.

SEGMENT TRANSFER TABLE Words

STT Length

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	U	0	0	0	0	0	0	LENGTH							

U Uncallable bit for STT# = 0(enter at PB + 0)
LENGTH Maximum = 255 (Calls from external segments may reference only the first 128 entries, PL thru PL-127.)

Local Program Label

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	U	ADDRESS													

U Uncallable bit
ADDRESS PB relative, + only

External Program Label

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	STT #									SEG #					

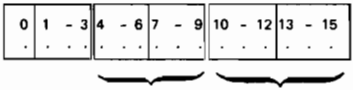
STT # = STT = entry number in target segment, maximum = 127
SEG # = Target segment

INTERRUPTS/TRAPS

STT Entry Number	Interrupt	Parameter
	EXTERNAL INTERRUPT	DEV#
1	BOUNDS VIOLATION	100401
2	ILLEGAL MEMORY ADDRESS	101001
3	NON-RESPONDING MODULE	101401
4	SYSTEM PARITY ERROR	102001
5	ADDRESS PARITY ERROR	102401
6	DATA PARITY ERROR	103001
7	MODULE INTERRUPT	MODULE#
11	POWER FAIL	104401
20	UNIMPLEMENTED INSTRUCTION	110001
21	STT VIOLATION	110401
22	CST VIOLATION	111001
23	DST VIOLATION	111401
24	STACK UNDERFLOW	112001
25	PRIVILEGED MODE VIOLATION	112401
30	STACK OVERFLOW	114001
31	USER TRAPS	
	INTEGER OVERFLOW	1
	FLOATING-POINT OVERFLOW	2
	FLOATING-POINT UNDERFLOW	3
	INTEGER DIVIDE BY 0	4
	FLOATING-POINT DIVIDE BY 0	5
	EXT. PRECISION OVERFLOW	10
	EXT. PRECISION UNDERFLOW	11
	EXT. PRECISION DIVIDE BY 0	12
	DECIMAL OVERFLOW	13
	INVALID ASCII DIGIT	14
	INVALID DECIMAL DIGIT	15
	INVALID SOURCE WORD COUNT	16
	RESULT WORD COUNT OVERFLOW	17
	DECIMAL DIVIDE BY 0	20
37	ABSENT CODE SEGMENT	
	PCAL	P-LABEL
	EXIT	N
	IXIT	0
40	TRACE	
	PCAL	P-LABEL
	EXIT	N
	IXIT	0
41	STT ENTRY UNCALLABLE	P-LABEL
42	ABSENT DATA SEGEMNT	DST #
43	POWER ON	121401
44	COLD LOAD	0

NOTE: If parameter not shown, parameter is external program label.

00 STACK OPS



CC = condition, O = overflow, C = carry, CI = clears bit

C C	O	C			C C	O	C		
			00	NOP				40	DEL
			01	DELB				41	ZROB
			02	DDEL	A			42	LDXB
			03	ZROX	A			43	STAX
A	X	X	04	INCX	A			44	LDXA
A	X	X	05	DECX	A			45	DUP
			06	ZERO	A			46	DDUP
			07	DZRO	A			47	FLT
C			10	DCMP	C			50	FCMP
A	X	X	11	DADD	A	X		51	FADD
A	X	X	12	DSUB	A	X		52	FSUB
A	CI	X	13	MPYL	A	X		53	FMPY
A	X		14	DIVL	A	X		54	FDIV
A	X		15	DNEG	A			55	FNEG
A			16	DXCH	A			56	CAB
C			17	CMP	C			57	LCMP
A	X	X	20	ADD	A		X	60	LADD
A	X	X	21	SUB	A		X	61	LSUB
A	X		22	MPY	A		X	62	LMPY
A	X		23	DIV	A	X		63	LDIV
A	X	X	24	NEG	A			64	NOT
A			25	TEST	A			65	OR
A			26	STBX	A			66	XOR
A		X	27	DTST	A			67	AND
A			30	DFLT	A	X	X	70	FIXR
B			31	BTST	A	X	X	71	FIXT
A			32	XCH				72	SPARE
A	X	X	33	INCA	A	X	X	73	INCB
A	X	X	34	DECA	A	X	X	74	DECB
A			35	XAX				75	XBX
A	X	X	36	ADAX	A	X	X	76	ADBX
A	X	X	37	ADXA	A	X	X	77	ADXB

01 SHIFTS/BRANCHES

				0	1-3	4 - 6	7-9	10 - 12	13 - 15	
A				0	1	X	0 0 0	←SC→		ASL
A						X	0 0 1	←SC→		ASR
A						X	0 0 2	←SC→		LSL
A						X	0 0 3	←SC→		LSR
A						X	0 0 4	←SC→		CSL
A						X	0 0 5	←SC→		CSR
A						X	0 0 6	0	0	SCAN
A	X	X				X	0 0 7	←P branch→		IABZ
A						X	0 1 0	←SC→		TASL
A						X	0 1 1	←SC→		TASR
A	X	X				I	0 1 2	+/-→P branch→		IXBZ
A	X	X				I	0 1 3	+/-→P branch→		DXBZ
		CI				I	0 1 4	+/-→P branch→		BCY
A		CI				I	0 1 5	+/-→P branch→		BNCY
						X	0 1 6	0	0	TNSL
							1 7	0	0	SPARE
A						X	1 0 0	←SC→		DASL
A						X	1 0 1	←SC→		DASR
A						X	1 0 2	←SC→		DLSL
A						X	1 0 3	←SC→		DLSR
A						X	1 0 4	←SC→		DCSL
A						X	1 0 5	←SC→		DCSR
*						I	1 0 6	+/-→P branch→		CPRB§
A	X	X				I	1 0 7	+/-→P branch→		DABZ
	CI					I	1 1 0	+/-→P branch→		BOV
	CI					I	1 1 1	+/-→P branch→		BNOV
**						X	1 1 2	bit position→		TBC
**						X	1 1 3	bit position→		TRBC
**						X	1 1 4	bit position→		TSBC
**						X	1 1 5	bit position		TCBC
						I	1 1 6	+/-→P branch→		BRO
						I	1 1 7	+/-→P branch→		BRE
A	O	C				0	0 1 7	←SC→		QASR
A						1	0 1 7	←SC→		QASL

CI = clears bit
§ = user Index Reg.
** bit = 0 CCE, bit = 1 CCG or CCL
SC = shift count (0 -63)
P branch signed magnitude (0 - 31)
bit position (0 - 63) [MOD 16]
CPRB X <(S - 1)CCL,X>(S)CCG
(S-1) ≤X≤(5)CCE

02 MOVES/IMMEDIATES

0	1-3	4-6	7-9	10 - 12	13 15
.	. . .	. . .	. . .	. . .	. . .

			0 2	0	0	0PB/DB0	SDEC	MOVE
				0	0	1PB/DB0	SDEC	MVB
				0	1	0	SDEC	MVBL
				0	1	0	1	two-
				two-word "hardware" ins. See page 5.				
				0	1	1	SDEC	MABS
B		*		0	1	2	SDEC	SCW
				0	1	3	SDEC	MTDS
				0	1	4	SDEC	MVLB
				0	1	5	SDEC	MDS
		*		0	1	6	SDEC	SCU
				0	1	7	SDEC	MFDS
B				0	2	ONA U	SDEC	MVBW
C				0	2	1 PB/DB0	SDEC	CMPB
				0	3	0	0	RSW
A				0	3	0	1	LLSH§
			0 2	0	3	0	2	two-
				two-word I/O ins. See page 5.				
A				0	3	2	0	PLDA§
				0	3	2	1	PSTA§
A				0	3	4	0	LSEA
				0	3	4	1	SSEA
A				0	3	4	2	LDEA
				0	3	4	3	SDEA
				0	3	6	0	IXIT
				0	3	6	1	NOP
				0	3	6	2	PCN
				0	3	6	3	NOP
A	X			0	4	1	0	EADD
A	X			0	4	1	1	ESUB
A	X			0	4	1	2	EMPY
A	X			0	4	1	3	EDIV
A				0	4	1	4	ENEG
C				0	4	1	5	ECMP
				0	5	7	0	DMUL
				0	5	7	1	DDIV
A		X		0	6	0	1	DMPY
C	O	C						
C								

02 MOVES/IMMEDIATES (cont.)

			0	1-3	4-6	7-9	10-12	13-15	
A	X				0	6	S0	2	CVAD
A	X				0	6	S0	3	CVDA
A	X				0	6	S0	4	CVBD
A	X		0	2	0	6	S0	5	CVDB
A	X	X			0	6	S S0	6	SLD
A	X	X			0	6	S S0	7	NSLD
A	X				0	6	S S1	0	SRD
A	X				0	6	S S1	1	ADDD
C	X				0	6	S S1	2	CMPD
A	X				0	6	S S1	3	SUBD
A	X				0	6	S 1	4	MPYD
A					1	0	← Imm Opr →		LDI
					1	1	← Imm Opr →		LDXI
C					2	0	← Imm Opr →		CMPI
A	X	X			2	1	← Imm Opr →		ADDI
A	X	X			3	0	← Imm Opr →		SUBI
A	X				3	1	← Imm Opr →		MPYI
A					4	0	← Imm Opr →		DIVI
					4	1	SBK DB DL Z S TA X Q S		PSHR
A					5	0	← Imm Opr →		LDNI
					5	1	← Imm Opr →		LDXN
C					6	0	← Imm Opr →		CMPN
A					6	1	J J J K K K		EXF
A					7	0	J J J K K K		DPF
					7	1	SBK DB DL Z S TA X Q S		SETR
C	O	C							
C									

S / SS } SDEC

§ = Uses Index Register

SDEC pop stack (0 -3)

Imm Opr Immediate operand (0 -255)

JJJJ Beginning bit position (0 -15)

KKKK field length (0 -15)

* SCW CARRY → Terminating character

* SCU CARRY → Terminating character

TWO-WORD "HARDWARE" INSTRUCTIONS

First Word	Second Word	MNE
020104	000000	RCCR
	000001	SCLR
	000002	TOFF
	000003	TON
	000004	SBM
	000007	Reserved
	000010	SINC

No Condition Codes Set

Note: The SBM instruction is not available on the Series 30/33.

TWO-WORD I/O INSTRUCTIONS

First Word	Second Word	MNE
020302	000000	SIOP
	1	HIOP
	2	RIOC
	3	WIOC
	6	INIT
	7	MCS
	10	SEML
	11	STRT
	12	DUMP

Instruction Codes 0-6 Set CC

Note: The SEML instruction is not available on the Series 44.

03 I/O, LINKAGE, CONTROL

0	1 - 3	4 - 6	7 - 9	10 - 12	13 - 15
.	. . .	. . .	. . .	. . .	. . .

			0 3						
				0	0	0	0 K K K		LST
				0	0	0	1 n n n n		PAUS
				0	0	1	0 0 0 0	D/E	SED
				0	0		6 0		XCHD
3				0	0		6 1		PSDB
				0	0		6 2		DISP
				0	0		6 3		PSEB
3				0	1		0 0		SMSK
				0	1		0 1		SCLK
				0	1		2 0		RMSK
				0	1		2 1		RCLK
*				0	1		1 0 K K K K		XEQ
				0	3		0 1 K K K K		SST
				0	3		1 1 n n n n		HALT
				0	1	← STT →			SCAL
				1	0	← STT →			PCAL
				1	1	← SDEC(+4) →			EXIT
				2	0	← SDEC(+1) →			SXIT
A				2	1	← Imm Opr →			ADXI
A				3	0	← Imm Opr →			SBXI
				3	1	← PL - Disp →			LLBL
A				4	0	← P + Disp →			LDPP
A				4	1	← P - Disp →			LDPN
				5	0	← Imm Opr →			ADDS
				5	1	← Imm Opr →			SUBS
				6	0	0	0		SPARE
A				6	1	← Imm Opr →			ORI
A				7	0	← Imm Opr →			XORI
A				7	1	← Imm Opr →			ANDI
C	O	C							
C									

KKKK Stack displacement (0 -15)
STT entry position (0 -255)
Imm Opr Immediate operand (0-255)
n Not used

1. normally CCE, non-responding CCL
D norm CCE, non-respond CCL, not ready CCG
3. norm CCE, IF error CCL
* XEQ depends on instruction executed.

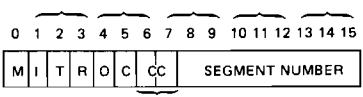
MEMORY REFERENCE

				01-34-67 - 9 10 - 12 13 - 15	
				. . . .	
A			0 4	X I 0 ← P →	LOAD
A				X I 1 ← DQS →	
			0 5	0 +/-←P rel branch→	TBA
				2 +/-←P rel branch→	MTBA
				4 +/-←P rel branch→	TBX
				6 +/-←P rel branch→	MTBX
				X I 1 DQS	STOR
C			0 6	X I 0 ← P →	CMPM
C				X I 1 ← DQS →	
A	X	X	0 7	X I 0 ← P →	ADDM
A	X	X		X I 1 ← DQS →	
A	X	X	1 0	X I 0 ← P →	SUBM
A	X	X		X I 1 ← DQS →	
A	X	X	1 1	X I 0 ← P →	MPYM
A	X	X		X I 1 ← DQS →	
A	X	X	1 2	X I 0 ← DQS →	INCM
A	X	X		X I 1 ← DQS →	DECM
A			1 3	X I 0 ← P →	LDX
A				X I 1 ← DQS →	
			1 4	X I 0 +/-←P rel branch→	BR
				X I 1 DQS indirect	BR
				I 0 1 GEL +/-P branch	BCC
B			1 5	X I 0 ← DQS →	LDB
A				X I 1 ← DQS →	LDD
			1 6	X I 0 ← DQS →	STB
				X I 1 ← DQS →	STD
			1 7	X I 0 ← P →	LRA
				X I 1 ← DQS →	
C					
C	O	C			

		7	8	9	
P	P+	0			0:377
	P-	1			0:377
DQS	DB+	0			0:377
	Q+	1	0		0:177
	Q-	1	1	0	0:77
	S-	1	1	1	0:77

GEL
1. Less than (BL)
2. Equal (BE)
3. Less than or equal (BLE)
4. Greater than (BG)
5. Not equal (BNE)
6. Greater or equal (BGE)

STATUS REGISTER



M Mode User Privileged
I Ext Interrupts Enabled
T User Traps Enabled
R Right Stack Op Pending
O Overflow
C Carry
CC Condition Code
CCL = 1
CCE = 2
CCG = 0

CONDITION CODES

CCA CCL Operand <0
CCE Operand = 0
CCG Operand >0

CCB CCL Special ASCII Char
CCE Alphabetic
CCG Numeric

CCC CCL Operand 1 $<$ OPR 2
CCE Operand 1 = OPR 2
CCG Operand 1 $>$ OPR 2

CC FIELD

6-7	Condition
0	$>$G
1	$<$L
2	= E
3	Undefined

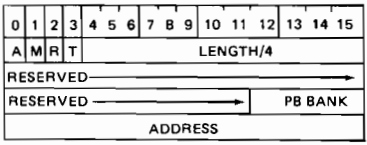
STACK MARKER (normal)

	X INDEX
	Relative P P + 1 – PB
	STATUS
Q,S	ΔQ

SPECIAL STACK MARKER (resulting from ICS interrupt)

	X INDEX
	Relative P P + 1 –PB
	STATUS
Q	ΔQ
	DB BANK
S	DB

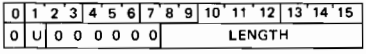
CODE SEGMENT TABLE



A Absence bit (=1 if segment is absent)
M Mode bit (=1 if privileged mode)
R Reference bit (for statistical use by operating system, set to 1 when accessed)
T Trace bit (=1 to call Trace routine)
LENGTH This value times 4 (max = 16,380) = segment length
ADDRESS Absolute memory address (for PB) or low order 16 bits of absolute disc address if absent.
PB BANK Bank number if present or High Order Disc Address if absent.

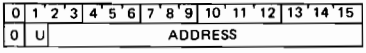
SEGMENT TRANSFER TABLE Words

STT Length



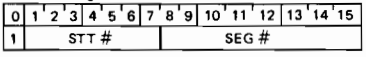
U Uncallable bit
LENGTH Maximum = 255 (Calls from external segments may reference only the first 128 entries, PL thru PL-127.)

Local Program Label



U Uncallable bit
ADDRESS PB relative, + only

External Program Label



STT # = STT entry number in target segment, maximum = 127
SEG # = Target segment

INTERRUPTS/TRAPS

STT Entry Number	Interrupt	Parameter
	EXTERNAL INTERRUPT	DEV#
1	BOUNDS VIOLATION	100401
3	NON-RESPONDING MODULE	101401
6	MEMORY DATA PARITY ERROR	103001
11	POWER FAIL	104401
12	SYSTEM CLOCK	(CR-LC)
20	UNIMPLEMENTED INSTRUCTION	110001
21	STT VIOLATION	110401
22	CST VIOLATION	111001
23	DST VIOLATION	111401
24	STACK UNDERFLOW	112001
25	PRIVILEGED MODE VIOLATION	112401
30	STACK OVERFLOW	114001
31	USER TRAPS	114401
	INTEGER OVERFLOW	1
	FLOATING-POINT OVERFLOW	2
	FLOATING-POINT UNDERFLOW	3
	INTEGER DIVIDE BY 0	4
	FLOATING-POINT DIVIDE BY 0	5
	EXT. PRECISION OVERFLOW	10
	EXT. PRECISION UNDERFLOW	11
	EXT. PRECISION DIVIDE BY 0	12
	DECIMAL OVERFLOW	13
	INVALID ASCII DIGIT	14
	INVALID DECIMAL DIGIT	15
	INVALID SOURCE WORD COUNT	16
	RESULT WORD COUNT OVERFLOW	17
	DECIMAL DIVIDE BY 0	20
37	ABSENT CODE SEGMENT	
	PCAL	P-LABEL
	EXIT	N
	IXIT	0
40	TRACE	
	PCAL	P-LABEL
	EXIT	N
	IXIT	0
41	STT ENTRY UNCALLABLE	P-LABEL
42	ABSENT DATA SEGEMNT	DST #
43	POWER ON	121401
44	COLD LOAD	CHAN-DEV #

NOTE: If parameter not shown, parameter is external program label.

Special Keys and Codes

Special Terminal Keys

Key	Meaning
X^C	Delete current line
H^C	Delete last character
Y^C	Subsystem "break"
Q^C	Place term in tape mode
J^C	Turn on linefeeds
M^C	Produces a "RETURN" (Carriage return and line feed)
BREAK	MPE/3000 break
ESC;	Stop echoing
ESC:	Resume echoing
F^C	Terminate termtype = 10 hang
HP 264X Terminals	
S^C	Stop output
Q^C	Resume output
$[^C$	ESC

029 Card-Punch Transliterations

ASCII	[= 029¢
ASCII	\ = 029 0-8-2
ASCII	] = 029 !
ASCII	∧ = 029 ¬
ASCII	! = 029 !

ASCII Character Substitutes

← is _

↑ is ∧

[is shift K

] is shift M

Part No. 30000-90049
Printed in U.S.A. 1/81

Incorporated 3/19/82

MANUAL UPDATE

Replaced pages follow

MANUAL IDENTIFICATION	UPDATE IDENTIFICATION
Part Number: 30000-90049	Update Number: 1
Print Date: JAN 1981	Print Date: APR 1981

Title: MPE SOFTWARE POCKET GUIDE

THE PURPOSE OF THIS MANUAL UPDATE
is to accumulate all changes to the current edition of the manual. Earlier updates, if any, are contained herein. This update consists of: this cover letter, a revised "List of Effective Pages," and all new and changed pages (backup pages are provided when necessary).

CHANGED PAGES ARE IDENTIFIED
by the date of the update at the bottom of the page and a vertical bar in the outside margin opposite the changed material.

NEW PAGES ARE IDENTIFIED
by the date of the update at the bottom of the page. "New" pages are those which were not present in the original edition of the manual.

TO UPDATE YOUR MANUAL,
replace change pages with the update pages and insert any new pages. Destroy all replaced pages.

HEWLETT PACKARD

HEWLETT-PACKARD COMPANY
19447 PRUNERIDGE AVE., CUPERTINO, CALIFORNIA 95014

Incorporated 3/14/85

Replaced pages follow

LIST OF EFFECTIVE PAGES

Seventh Edition ..... Jan 1981

PRINTING HISTORY

Seventh Edition ..... Jan 1981

Section IV
IMAGE, QUERY

IMAGE
SCHEMA Processor
Operation 4-1
File Designators 4-1
Commands. 4-1
SCHEMA Structure 4-2
DBLOAD. 4-3
DBRECOV. 4-3
DBRESTOR. 4-4
DBSTORE. 4-4
DBUNLOAD 4-4
DBUTIL 4-5
Calling an IMAGE procedure. 4-7
Intrinsics Exceptional Conditions. 4-10
QUERY
Operation 4-12
Commands. 4-12
Statements. 4-16
Statement Parameters 4-17

Section V
KSAM, V/3000

KSAMUTIL
Operation 5-1
Commands. 5-1
SPL Intrinsics. 5-3
COBOL Procedures 5-10
BASIC Procedures. 5-12
FORTRAN Procedures. 5-14
V/3000
FORMSPEC. 5-15
REFSPEC 5-22
REFORMAT 5-24
COMAREA 5-27
RPG Interface. 5-29
ENTRY. 5-32

CONTENTS (Continued)

Section VI
UTILITIES

ASOCTABL to DISKED2 6-1
DPAN4 6-3
FREE2 to LISTDIR2 6-5
LISTLOG2 to MEMTIMER 6-6
MEMLOGAN 6-7
PATCH to SADUTIL 6-8
SLPATCH to SPOOK 6-11
VINIT 6-13

Section VII
SEGMENTER

Operation 7-1
Commands 7-1
Intrinsics 7-2

Section VIII
INTRINSICS

ACTIVATE to CLEANUSL 8-1
CLOCK to CREATEPROCESS 8-2
CTRANSLATE to DMOVIN 8-3
DMOVOUT to FCLOSE 8-4
FCONTROL 8-5
FDELETE to FFILEINFO 8-6
FGETINFO to FINDJCW 8-7
FLOCK to FREAD 8-8
FREADBACKWARD to FREADDIR 8-9
FREADLABEL to FSPACE 8-12
FUNLOCK to GET 8-13
GETDSEG to GETPROCINFO 8-14
GETUSERMODE to LOCKGLORIN 8-15
LOCKLOCRIN to PCONTROL 8-16
POPEN to PRINTFILEINFO 8-16a
PRINTOP to QUIT 8-17
QUITPROG to SETDUMP 8-18
SETJCW to UNLOCKLOCRIN 8-19
WHO to XCONTRAP 8-20
XLIBTRAP to ZSIZE 8-21

```
[ ;NOLABEL                                                    ]]]
[ ;LABEL=[volid] [,[type] [,[expdate][,seq]]                  ]]]

[;FORMS=formsmsg]

[ ;NOLOCK ]
[ ;LOCK   ]

[;DEV=[[dsdevise]#] [device] [,[outpri] [,numcopies]]]
[;CODE=[filecode]]
[;DISC=[numrec] [,[numextents] [,initalloc]]]
```

Defaults: for new files are:
recsize = 128 for DISC/TAPE
= -132 for LP
= -80 for CARD/PUNCH
= -72 for TERM
blockfactor = physical record/recsize
SHR (if ACC= IN, otherwise EXC)
BUF = 2
DEV = DISC
CODE = 0
DISC = 1023,8,1

Note: The parameter group [;DISC=[numrec] [,[numextents] [,initalloc]]] cannot be included if the parameter group

```
[=filereference] [ ,OLD     ] is specified.
                 [ ,OLDTEMP ]
```

For recsize, positive value indicates words and negative value indicates bytes.

For outputpriority, select a number between 1 (lowest priority) and 13 (highest priority) (Default is 1.)

For filecode see File System

User Pre-Defined (Back-referenced) Files
:FILE formaldesignator = *formaldesignator

Commands

For System-Defined Files:

```
:FILE          formaldesignator = $NULL

:FILE          formaldesignator = { $STDIN  }
                                  { $STDINX }
                                  { $STDLIST }

[;REC=[recsize]  [ ,[blockfactor]  [ [ F ] ,BINARY ] ] ]
                                    [ [ U ] ,ASCII  ]
                                    [ [ V ]         ]

[ ;CCTL   ]
[ ;NOCCTL ]

[;DEV=[[dsdevice]#] [device] [,[outpri] [,numcopies] ] ]

[ ;ACC= { IN      } ]
        { OUT     }
        { UPDATE  }
        { OUTKEEP }
        { APPEND  }
        { INOUT   }

[ ;NOBUF              ]
[ ;BUF [=numbuffers] ]

[ ;EXC ]
[ ;EAR ]
[ ;SHR ]

[ ;MR   ]
[ ;NOMR ]

[ ;WAIT   ]
[ ;NOWAIT ]

[ ;NOLABEL                                                    ]
[ ;LABEL=[volid] [,[type] [,[exdate] [,seq] ] ] ]

[;FORMS=formsmsg]

[ ;NOLOCK ]
[ ;LOCK   ]
```

Spooling Command/Event Matrix

Console Commands

System Start Up

Type	From	Thumb-wheel	Sw Reg	Effect:
WARM*	DISC	WARM	%0004 *	Permits recovery of incompletely processed spooled jobs and spoolfiles.
COOL	DISC	WARM	%0004 *	Standard operation. All permanent user files are saved. All temporary files, jobs, and sessions done before COOLSTART are lost.
COLD	SERIAL STORAGE DEVICE	COLD	%3006	System files and I/O configuration come from tape; user files, directory, accounting information, and global RINs are retained – obtained from disc.
UPDATE	SERIAL STORAGE DEVICE	COLD	%3006	System files come from tape; I/O configuration, directory, accounting information, and global RINs remain unchanged.
RELOAD	SERIAL STORAGE DEVICE	COLD	%3006	Complete MPE comes from tape – system files, I/O configuration, file directory, and user files. If the disc label is good, some items are not reloaded. To insure that the entire system is restored, follow the RELOAD with an update option.

*All cold-load options except WARM allow the operator to alter the input/output device configuration currently in effect.

Note: Thumbwheels apply to Series 30/33/44; Switch Registers apply to Series II/III.

*Value should be the DRT number of the system disc.

```
                    IV          IV          L
FCONTROL    (filenum, controlcode, param);
```

Values for controlcode:

0 = General.
1 = LP.
2 = Complete all I/O.
3 = Param:=status.
4 = Set timeout.
5 = Rewind.
6 = Write EOF.
7 = Forward to tape mark.
8 = Back to tape mark.
9 = Rewind/offline.
10 = Change terminal input speed.
11 = Change terminal output speed.
12 = Turn echo facility on.
13 = Turn echo facility off.
14 = Disable the system break function.
15 = Enable the system break function.
16 = Disable the subsystem break function.
17 = Enable the subsystem break function.
18 = Disable tape mode option.
19 = Enable tape mode option.
20 = Disable the terminal input time.
21 = Enable the terminal input timer.
22 = Read the terminal input timer.
23 = Disable parity checking.
24 = Enable parity checking.
25 = Define line-termination characters for terminal input.
26 = Disable binary transfers.
27 = Enable binary transfers.
28 = Disable user block mode transfers.
29 = Enable user block mode transfers.
34 = Disable line deletion echo suppression.
35 = Enable line deletion echo suppression.
36 = Set parity.
37 = Allocate a terminal.
38 = Set terminal type.
39 = Obtain terminal type information.
40 = Obtain terminal output speed.
41 = Set unedited terminal mode.
43 = Abort pending NO-WAIT I/O request.
45 = Enable/Disable extended wait.
46 = Enable/Disable reading writer's ID.
47 = Nondestructive read.

Condition Codes: CCE, CCL

```
                    IV        DV      O-V
FDELETE        (filenum, recnum);
        Condition Codes:  CCE, CCG, CCL

                    I        LA        I
FERRMSG        (errorcode, msgbuf, msglgth);
        Condition Codes:  CCE, CCL, CCG

                    IV        IV          BA
FFILEINFO      (filenum [,itemnum1, itemvalue1]
                        [,itemnum2, itemvalue2]
                        [,itemnum3, itemvalue3]
                        [,itemnum4, itemvalue4]
                        [,itemnum5, itemvalue5]);
```

ITEM NO.	TYPE	ITEM	UNITS
1	BA	filename (see FGETINFO)	
2	L	foptions (see FGETINFO)	
3	L	aoptions (see FGETINFO)	
4	I	recsize (see FGETINFO)	words/bytes
5	I	devtype (see FGETINFO)	
6	L	ldnum (see FGETINFO)	
7	L	hdaddr (see FGETINFO)	
8	I	filecode (see FGETINFO)	
9	D	recpt (see FGETINFO)	
10	D	eof (see FGETINFO)	
11	D	flimit (see FGETINFO)	records
12	D	logcount (see FGETINFO)	records
13	D	physcount (see FGETINFO)	records
14	I	blksize (see FGETINFO)	words/bytes
15	L	extsize (see FGETINFO)	sectors
16	I	numextents (see FGETINFO)	
17	I	userlabels (see FBETINFO)	
18	BA	creatorid (see FGETINFO)	
19	D	labaddr (see FGETINFO)	
20	I	blocking factor (See FOPEN)	
21	I	physical block size	words
22	I	data block size	words
23	I	offset to data in blocks	words
24	I	offset to Active Record Table in block (R10 files)	words
25	I	size of Active Record Table	words
26	BA	vol. ID (label tape) (see Label Tapes)	
27	BA	vol. set ID (label tape) (see Label Tapes)	

28	I	expiration date (CALENDAR format (see Label Tapes)
29	I	file sequence number (see Label Tapes)
30	I	reel number (see Label Tapes)
31	I	sequence type (see Label Tapes)
32	I	creation date (CALENDAR Format (see Label Tapes)
33	I	label type (see Label Tapes)
34		RESERVED
35		
36	L	File Allocation Date (CALENDAR format)
37	D	File Allocation Time (CLOCK format)
38	L	SPOOFLE Device File Number (#O or #I number) (see File Code)
40	D	disc or diskette device status
41	I	device type
42	I	device subtype

Note: Parameters must appear in pairs.

Condition Codes: CCE, CCL

```
                    IV      BA      L       L       I
FGETINFO    (filenum, filename, foptions, aoptions, recsize,

                    I       L       L       I       D   D
devtype, ldnum, hdaddr, filecode, recpt, eof,

                    D       D       D       I       L
flimit, logcount, physcount, blksize, extsize,

                    I       I       BA      D       O-V
numextents, userlabels, creatorid, labaddr);
```

Note: filename parm must be 28 bytes long.

Condition Codes: CCE, CCL

```
                    BA      L       I
FINDJCW     (jcwname, jcwvalue, status);
```

Condition Codes: None
(status <> 0 is error)

```
                    IV       LV
FLOCK          (filenum, lockcond);
```

Condition Codes: CCE, CCG, CCL

```
                  LV    BA
FMTCALENDAR (date, string);
```

Condition Codes: None

```
             DV    BA
FMTCLOCK (time, string);
```

Condition Codes: None

```
                  LV   DV    BA
FMTDATE        (date, time, string);
```

Condition Codes: None

FOPEN

```
  I                          BA          LV        LV       IV
filenum: = FOPEN (formaldesignator, foptions, aoptions, recsize,

          BA       BA         IV          IV              IV
        device, formmsg, userlabels, blockfactor, numbuffers,

          DV       IV           IV         IV       O-V
        filesize, numextents, initialloc, filecode);
```

<<returns "Filenum' used in other intrinsics>>

Condition Codes: CCE, CCL

```
                    IV       DV
FPOINT         (filenum, recnum);
```

<<point to a record directly>>

Condition Codes: CCE, CCG, CCL

```
                I                  IV      LA      IV
FREAD          lgth: = FREAD (filenum, target, tcount);
```

Condition Codes: CCE, CCG, CCL

```
                    I                                  IV
FREADBACKWARD  lgth: = FREADBACKWARD (filenum,
                          LA     IV
                       target, tcount);
```

Condition Codes: CCE, CCG, CCL

```
                  IV      LA     IV      DV
FREADDIR      (filenum, target, tcount, recnum);
```

Condition Codes: CCE, CCG, CCL

FOPTIONS:

BITS	(0:2)	(2:3)		(5:1)	(6:1)	(7:1)	(8:2)	(10:3)	(13:1)	(14:2)
FIELD	Reserved	File Type		Disallow :FILE	MPE Tape Labels	Carriage Control	Record Format	Default Designator	ASCII/ Binary	Domain
MEANING		00	0≡STD	0≡Allow :FILE	0≡NON LABEL-ED TAPE	0≡NOCCTL	00≡Fixed	000≡filename	0≡Binary	00≡New file
		00	1≡KSAM	1≡No :FILE	1≡ LABEL-ED TAPE	1≡CCTL	01≡Variable	001≡$STDLIST	1≡ASCII	01≡Old Permanent File
		01	0≡RIO				10≡Unde-fined	010≡$NEWPASS		
		10	0≡CIR					011≡$OLDPASS		10≡Old Temporary File
		11	0≡MSG					100≡$STDIN		
								101≡$STDINX		11≡Old Perm. or Temp. File
								110≡$NULL		

BITS	(0:3)	(3:1)	(4:1)	(5:2)	(7:1)	(8:2)	(10:1)	(11:1)	(12:4)	
FIELD	Reserved	File Copy	No-Wait I/O	Multi Access	Inhibit Buffering	Exclusive Access	Dynamic Locking	Multi-record Access	Access Type	
MEANING		0≡access in file's netive mode 1≡access as standard sequential file	1≡No Wait 2≡Non No-Wait	00≡Non-multi-access 01≡Only Intra-job multi-access 10≡Inter-job multi-access allowed	0≡BUF 1≡NOBUF	00≡Default 01≡Exclusive 10≡Exclusive Access Read 11≡Share	0≡No FLOCK Allowed 1≡ FLOCK Allowed	0≡No Multi-Record 1≡Multi record	0	000≡Read only
									0	001≡Write only
									0	010≡Write (save) only
									0	011≡Append only
									0	100≡Read/write
									0	101≡Update
									0	110≡Execute

```
                    IV      LA      IV      IV      O-V
FREADLABEL  (filenum, target, tcount, labelid);
```

Condition Codes: CCE, CCG, CCL

```
                    IV      DV
FREADSEEK   (filenum, recnum);
```

Condition Codes: CCE, CCG, CCL

```
                    LV  LV
FREEDSEG    (index, id);
```

Condition Codes: CCE, CCG, CCL

```
FREELOCRIN;
```

Condition Codes: CCE, CCG, CCL

```
                    L                           IV      IV
FRELATE     intordup: = FRELATE (infilenum, listfilenum);
```

Condition Codes: CCE, CCG, CCL

```
                    IV          BA
FRENAME     (filenum, newfilereference);
```

Condition Codes: CCE, CCL

```
                    IV          LV
FSETMODE    (filenum, modeflags);
```

Note: Modeflags = (14:1) ← (0: = unblocked I/O, w - o wait | 1≡complete are I/O)
(13:1) ← (0: = automatic CR/LF | 1≡suppress CR/LF)
(12:1) ← (0: = CCE for tape errors | 1≡CCL for tape errors)

Condition Codes: CCE, CCL

```
                    IV          IV
FSPACE      (filenum, displacement);
```

<<+≡forward, -≡backward>>

Condition Codes: CCE, CCG, CCL

```
                    IV
FUNLOCK        (filenum);
```

Condition Codes: CCE, CCG, CCL

```
                    IV      LA      IV
FUPDATE        (filenum, target, tcount);
```

Condition Codes: CCE, CCG, CCL

```
                    IV      LA      IV      LV
FWRITE         (filenum, target, tcount, control);
```

Note: Control: =(0≡normal | 1≡ use 1st character of "target" | "1"≡ page eject | %320 ≡ no cr, no LF)

Condition Codes: CCE, CCG, CCL

```
                    IV      LA      IV      DV
FWRITEDIR      (filenum, target, tcount, recnum);
```

Condition Codes: CCE, CCG, CCL

```
                    IV      LA      IV      IV      O-V
FWRITELABEL    (filenum, target, tcount, labelid);
```

Condition Codes: CCE, CCG, CCL

GENMESSAGE

```
  I                          IV       IV       IV      BA      IV
msglen: = GENMESSAGE (filenum, setnum, msgnum, buff, buffsize,

          LV      LV      LV      LV      LV      LV      IV
        parmask, parm1, parm2, parm3, parm4, parm5, msgdest,

          I       O-V
        errnum);
```

Condition Codes: CCE, CCG, CCL

```
                    I              IA I      I
GET             ifun: = GET(itag,il,ionumber)
```

Intrinsics

```
                    L     I    LV
GETDSEG         (index, length, id);
```

Condition Codes: CCE, CCG, CCL

```
                   L
GETJCW          jcw: =GETJCW;
```

Condition Codes: None

```
                   LV
GETLOCRIN       (rincount);
```

Condition Codes: CCE, CCG, CCL

```
                    I
GETORIGIN       source: = GETORIGIN;
```

Condition Codes: None

```
                  IV      LV       IV    O-V
GETPRIORITY     (pin, priorityclass, rank);
```

Condition Codes: CCE, CCG, CCL

```
GETPRIVMODE; O-P
```

Condition Codes: CCE, CCG

```
                    I                      IV
GETPROCID       pin: = GETPROCID (numson);
```

Condition Codes: None

```
                   D                       IV
GETPROCINFO     statinfo: = GETPROCINFO (pin);
```

Condition Codes: CCE, CCG, CCL

RUN TIME ERRORS

MSGNO	Message
0	END OF FILE
1	ILLEGAL DB REGISTER
2	ILLEGAL CAPABILITY
3	OMITTED PARAMETER
4	INCORRECT S REGISTER
5	PARAMETER ADDRESS VIOLATION
6	PARAMETER END ADDRESS VIOLATION
7	ILLEGAL PARAMETER
8	PARAMETER VALUE INVALID
9	INCORRECT Q REGISTER

FILE SYSTEM ERRORS

0	END OF FILE (FSERR 0)
1	ILLEGAL DB REGISTER SETTING (FSERR 1)
2	INVALID OPTIONS
8	ILLEGAL PARAMETER VALUE (FSERR 8)
9	INVALID FILE TYPE SPECIFIED IN FOPTIONS (FSERR 9)
10	INVALID RECORD SIZE SPECIFICATION
11	INVALID BLOCKSIZE
20	INVALID OPERATION (FSERR 20)
21	DATA PARITY ERROR (FSERR 21)
22	SOFTWARE TIME-OUT (FSERR 22)
23	END OF TAPE (FSERR 23)
24	UNIT NOT READY (FSERR 24)
25	NO WRITE-RING ON TAPE (FSERR 25)
26	TRANSMISSION ERROR (FSERR 26)
27	I/O TIME-OUT (FSERR 27)
28	TIMING ERROR OR DATA OVERRUN (FSERR 28)
29	SIO FAILURE (FSERR 29)
30	UNIT FAILURE (FSERR 30)
31	END OF LINE (FSERR 31)
32	SOFTWARE ABORT (FSERR 32)
33	DATA LOST (FSERR 33)
34	UNIT NOT ON-LINE (FSERR 34)
35	DATA-SET NOT READY (FSERR 35)
36	INVALID DISC ADDRESS (FSERR 36)
37	INVALID MEMORY ADDRESS (FSERR 37)
38	TAPE PARITY ERROR (FSERR 38)
39	RECOVERED TAPE ERROR (FSERR 39)
40	OPERATION INCONSISTENT WITH ACCESS TYPE (FSERR 40)

41	OPERATION INCONSISTENT WITH RECORD TYPE (FSERR 41)
42	OPERATION INCONSISTENT WITH DEVICE TYPE (FSERR 42)
43	WRITE EXCEEDS RECORD SIZE (FSERR 43)
44	UPDATE AT RECORD ZERO (FSERR 44)
45	PRIVILEGED FILE VIOLATION (FSERR 45)
46	OUT OF DISC SPACE (FSERR 46)
47	I/O ERROR ON FILE LABEL (FSERR 47)
48	INVALID OPERATION DUE TO MULTIPLE FILE ACCESS (FSERR 48)
49	UNIMPLEMENTED FUNCTION (FSERR 49)
50	NONEXISTENT ACCOUNT (FSERR 50)
51	NONEXISTENT GROUP (FSERR 51)
52	NONEXISTENT PERMANENT FILE (FSERR 52)
53	NONEXISTENT TEMPORARY FILE (FSERR 53)
54	INVALID FILE REFERENCE (FSERR 54)
55	DEVICE UNAVAILABLE (FSERR 55)
56	INVALID DEVICE SPECIFICATION (FSERR 56)
57	OUT OF VIRTUAL MEMORY (FSERR 57)
58	NO PASSED FILE (FSERR 58)
59	STANDARD LABEL VIOLATION (FSERR 59)
60	GLOBAL RIN UNAVAILABLE (FSERR 60)
61	OUT OF GROUP DISC SPACE (FSERR 61)
62	OUT OF ACCOUNT DISC SPACE (FSERR 62)
63	USER LACKS NON-SHARABLE DEVICE CAPABILITY (FSERR 63)
64	USER LACKS MULTI-RIN CAPABILITY (FSERR 64)
65	PUNCH HOPPER EMPTY (FSERR 65)

66	PLOTTER LIMIT SWITCH REACHED (FSERR 66)
67	PAPER TAPE ERROR (FSERR 47)
68	INSUFFICIENT SYSTEM RESOURCES (FSERR 68)
69	I/O ERROR (FSERR 69)
70	I/O ERROR WHILE PRINTING HEADER/ TRAILER (FSERR 70)
71	TOO MANY FILES OPEN (FSERR 71)
72	INVALID FILE NUMBER (FSERR 72)
73	BOUNDS VIOLATION (FSERR 73)
77	NO-WAIT I/O PENDING (FSERR 77)
78	NO NO-WAIT I/O PENDING FOR ANY FILE (FSERR 78)
79	NO NO-WAIT I/O PENDING FOR SPECIAL FILE (FSERR 79)
80	SPOOFLE SIZE EXCEEDS CONFIGURATION (FSERR 80)
81	NO "SPOOL" CLASS IN SYSTEM (FSERR 81)
82	INSUFFICIENT SPACE FOR SPOOFLE (FSERR 82)
83	I/O ERROR ON SPOOFLE (FSERR 83)
84	DEVICE UNAVAILABLE FOR SPOOFLE (FSERR 84)
85	OPERATION INCONSISTENT WITH SPOOLING (FSERR 85)
86	NONEXISTENT SPOOFLE (FSERR 86)
87	BAD SPOOFLE BLOCK (FSERR 87)
88	SPOOLING ERROR (FSERR 88)
89	POWER FAILURE (FSERR 89)
90	EXCLUSIVE VIOLATION: FILE BEING ACCESSED (FSERR 90)
91	EXCLUSIVE VIOLATION: FILE ACCESSED EXCLUSIVELY (FSERR 91)
92	LOCKWORD VIOLATION (FSERR 92)
93	SECURITY VIOLATION (FSERR 93)
94	USER IS NOT CREATOR (FSERR 94)
95	READ COMPLETED DUE TO BREAK (FSERR 95)
96	DISC I/O ERROR (FSERR 96)
97	NO CONTROL Y PIN (FSERR 97)
98	READ TIME OVERFLOW (FSERR 98)
99	EOT AND BACKSPACE TAPE (FSERR 99)

100	DUPLICATE PERMANENT FILE NAME (FSERR 100)
101	DUPLICATE TEMPORARY FILE NAME (FSERR 101)
102	I/O ERROR ON DIRECTORY (FSERR 102)
103	PERMANENT FILE DIRECTORY OVERFLOW (FSERR 103)
104	TEMPORARY FILE DIRECTORY OVERFLOW (FSERR 104)
105	BAD VARIABLE BLOCK STRUCURE (FSERR 105)
106	EXTENT SIZE EXCEEDS MAXIMUM (FSERR 106)
107	INSUFFICIENT SPACE FOR USER LABELS (FSERR 107)
108	INVALID FILE LABEL (FSERR 108)
109	INVALID CARRIAGE CONTROL (FSERR 109)
110	ATTEMPT TO SAVE PERMANENT FILE AS TEMPORARY (FSERR 110)
111	USER LACKS SAVE FILES (SF) CAPABILITY (FSERR 111)
112	USER LACKS PRIVATE VOLUMES (UV) CAPABILITY (FSERR 112)
113	VOLUME SET NOT MOUNTED – MOUNT PROBLEM (FSERR 113)
114	VOLUME SET NOT DISMOUNTED – DISMOUNT PROBLEM (FSERR 114)
115	ATTEMPTED RENAME ACROSS VOLUME SETS – REJECTED (FSERR 115)
116	INVALID TAPE LABEL FOPEN PARAMETERS (FSERR 116)
117	ATTEMPT TO WRITE ON AN UNEXPIRED TAPE FILE (FSERR 117)
118	INVALID HEADER OR TRAILER TAPE LABEL (FSERR 118)
119	I/O ERROR POSITIONING TAPE FOR TAPE LABELS (FSERR 119)
120	ATTEMPT TO WRITE IBM STANDARD TAPE LABEL (FSERR 120)
121	TAPE LABEL LOCKWORD VIOLATION (FSERR 121)

123	END OF TAPE VOLUME SET (FSERR 123)
124	ATTEMPT TO APPEND LABELED TAPE (FSERR 124)
137	DEFECTIVE TRACK ON FOREIGN DISC (FSERR 137)
138	TRACK DOES NOT EXIST ON FOREIGN DISC (FSERR 138)
139	DELETED RECORD ON IBM DISKETTE (FSERR 139)
148	INACTIVE RIO RECORD (FSERR 148)
149	MISSING ITEM NUMBER OR RETURN-VARIABLE (FSERR 149)
150	INVALID ITEM NUMBER (FSERR 150)
151	CURRENT RECORD WAS LAST RECORD WRITTEN BEFORE SYSTEM CRASHED.
152	UNRECOGNIZED KEYWORD IN FOPEN DEVICE PARAMETER (FSERR 152)
153	EXPECTED ";" OR CARRIAGE RETURN IN DEVICE PARAMETER (FSERR 153)
154	ENVIRONMENT FILE OPEN ERROR (FSERR 154)
155	FILECODE NOT THAT OF ENVIRONMENT FILE (FSERR 155)
156	ENVIRONMENT HEADER RECORD INCORRECT (FSERR 156)
157	UNCOMPILED ENVIRONMENT FILE (FSERR 157)
158	ERROR READING ENVIRONMENT FILE (FSERR 158)
159	ERROR CLOSING ENVIRONMENT FILE (FSERR 159)
160	ERROR DOING FDEVICECONTROL FROM ENVIRONMENT (FSERR 160)
161	TOO MANY PARAMETERS IN DEVICE STRING – OVERFLOW (FSERR 161)
162	EXPECTED "=" AFTER KEYWORD IN DEVICE PARAMETER (FSERR 162)
171	DUPLICATE KEY VALUE (FSERR 171)
172	NO SUCH KEY (FSERR 172)
173	TCOUNT PARAMETER LARGER THAN RECORD SIZE (FSERR 173)
174	CAN NOT GET EXTRA DATA SEGMENT (FSERR 174).

175	KSAN INTERNAL ERROR (FSERR 175)
176	ILLEGAL EXTRA DATA SEGMENT LENGTH (FSERR 176)
177	TOO MANY EXTRA DATA SEGMENTS FOR THIS PROCESS (FSERR 177)
178	EXTRA DATA SEGMENT TOO SMALL (FSERR 178)
179	THE FILE MUST BE LOCKED BEFORE ISSUING THIS INTRINSIC (FSERR 179)
181	INVALID KEY STARTING POSITION (FSERR 181)
182	FILE IS EMPTY (FSERR 182)
183	RECORD DOES NOT CONTAIN ALL KEYS (FSERR 183)
184	INVALID RECORD NUMBER (FFINDN INTRINSIC) (FSERR 184)
185	SEQUENCE ERROR IN PRIMARY KEY (FSERR 185)
186	INVALID KEY LENGTH – NUMERIC DISPLAY OR PACKED DECIMAL (FSERR 186)
187	INVALID KEY SPECIFICATION (FSERR 187)
188	INVALID DEVICE SPECIFICATION (FSERR 188)
189	INVALID RECORD FORMAT (FSERR 189)
190	INVALID KEY BLOCKING FACTOR VALUE (FSERR 190)
191	RECORD DOES NOT CONTAIN SEARCH KEY SPECIFIED FOR DELETION (FSERR 191)
192	SYSTEM FAILURE OCCURRED WHILE THE KSAM FILE WAS OPENED (FSERR 192)
201	INVALID ID SEQUENCE (FSERR 201)
202	INVALID TELEPHONE NUMBER (FSERR 202)
203	NO TELEPHONE LIST SPECIFIED (FSERR 203)
204	UNABLE TO ALLOCATE AN EXTRA DATA SEGMENT FOR DS/3000 (DSERR 204)
205	UNABLE TO EXPAND THE DS/3000 EXTRA DATA SEGMENT (DSERR 205)

206	SLAVE PTOP FUNCTION ISSUED FROM A MASTER PROGRAM. (DSERR 206)
207	SLAVE PTOP FUNCTION OUT OF SEQUENCE. (DSERR 207)
208	MASTER PTOP FUNCTION ISSUED BY A SLAVE PROGRAM. (DSERR 208)
209	SLAVE PROGRAM DOES NOT EXIST OR IS NOT PROGRAM FILE. (DSERR 209)
210	WARNING – INVALID MAXDATA OR DLSIZE FOR A SLAVE PROGRAM. SYSTEM DEFAULTS ARE IN EFFECT. (DSWARN 210)
211	SLAVE ISSUED A REJECT TO A MASTER PTOP OPERATION. (DSWARN 211)
212	FILE NUMBER RETURNED FROM IOWAIT IS NOT A DS LINE NUMBER (DSWARN 212)
213	EXCLUSIVE USE OF A DS LINE REQUIRES BOTH ND AND CS CAPABILITY (DSERR 213)
214	THE REQUESTED DS LINE HAS NOT BEEN OPEN WITH A USER :DSLINE COMMAND (DSERR 214)
215	DSLINE CANNOT BE ISSUED BACK TO THE MASTER COMPUTER. (DSERR 215)
216	MESSAGE REJECTED BY THE REMOTE COMPUTER (DSERR 216)
217	INSUFFICIENT AMOUNT OF USER STACK AVAILABLE (DSERR 217)
218	INVALID PTOP FUNCTION REQUESTED. (DSERR 218)
219	MULTIPLE POPEN. ONLY ONE MASTER PTOP OPERATION CAN BE ACTIVE ON A DS LINE. (DSERR 219)
220	PROGRAM EXECUTING GET WAS NOT CREATED BY POPEN. (DSERR 220)
221	INVALID DS MESSAGE FORMAT (INTERNAL DS ERROR) (DSERR 221)
222	MASTER PTOP FUNCTION ISSUED PRIOR TO A POPEN. (DSERR 222)
223	REQUEST TO SEND MORE DATA THAN SPECIFIED IN POPEN. (DSERR 223)
224	FILE EQUATIONS FOR A REMOTE FILE CONSTITUTE A LOOP. (DSERR 224)

225	CANNOT ISSUE POPEN TO A SLAVE SESSION IN BREAK MODE. (DSERR 225)
226	SLAVE PROGRAM HAS TERMINATED BEFORE EXECUTING "GET" (DSERR 226)
227	REMOTE HELLO MUST BE DONE TO INITIATE REMOTE SESSION. (DSERR 227)
236	COMMUNICATIONS HARDWARE HAS DETECTED AN ERROR. (DSERR 236)
237	CANNOT CURRENTLY GAIN ACCESS TO THE TRACE FILE. (DSERR 237)
238	COMMUNICATIONS INTERFACE ERROR. INTERNAL FAILURE. (DSERR 238)
239	COMMUNICATIONS INTERFACE ERROR. TRACE MALFUNCTION. (DSERR 239)
240	THE LOCAL COMMUNICATION LINE HAS NOT BEEN OPENED BY THE OPERATOR (DSERR 240)
241	THE DS LINE IS IN USE EXCLUSIVELY OR BY ANOTHER SUBSYSTEM (DSERR 241)
242	INTERNAL DS SOFTWARE MALFUNCTION (DSERR 242)
243	THE REMOTE COMPUTER IS NOT RESPONDING (DSERR 243)
244	COMMUNICATIONS INTERFACE ERROR. THE REMOTE COMPUTER RESET THE LINE (DSERR 244)
245	COMMUNICATIONS INTERFACE ERROR. RECEIVE TIMEOUT (DSERR 245)
246	COMMUNICATIONS INTERFACE ERROR. REMOTE HAS DISCONNECTED (DSERR 246)
247	COMMUNICATIONS INTERFACE ERROR. LOCAL TIME OUT (DSERR 247)
248	COMMUNICATIONS INTERFACE ERROR. CONNECT TIME OUT (DSERR 248)
249	COMMUNICATIONS INTERFACE ERROR. REMOTE REJECTED CONNECTION (DSERR 249)
250	COMMUNICATIONS INTERFACE ERROR. CARRIER LOST (DSERR 250)

251	COMMUNICATIONS INTERFACE ERROR. THE LOCAL DATA SET FOR THE DS LINE WENT NOT READY (DSERR 251).
252	COMMUNICATIONS INTERFACE ERROR. HARDWARE FAILURE (DSERR 252)
253	COMMUNICATIONS INTERFACE ERROR. NEGATIVE RESPONSE TO THE DIAL REQUEST BY THE OPERATOR (DSERR 253)
254	COMMUNICATIONS INTERFACE ERROR. INVALID I/O CONFIGURATION (DSERR 254)
255	COMMUNICATIONS INTERFACE ERROR. UNANTICIPATED ERROR CONDITION (DSERR 255)
300	NUMBER OF OPENS FOR FILE EXCEEDS 255. (FSERR 300)
301	FREE SPACE TABLE FOR LDEV ! IS FULL, RUN VINIT -COND