

ISIS-II POCKET REFERENCE

Order Number: 9800841-03

intel[®]

TABLE OF CONTENTS

	PAGE
Conventions	1
Device Filename Format	1
Control Characters	2
Command Syntax	2
ISIS-II System Calls	6
ISIS-II Error Messages	9
Hexadecimal-Decimal Conversion	11
Monitor Commands	12

CONVENTIONS:

UPPERCASE—must be entered as shown

<lower case>—variable information

[]—indicate optional field

...—field may be repeated

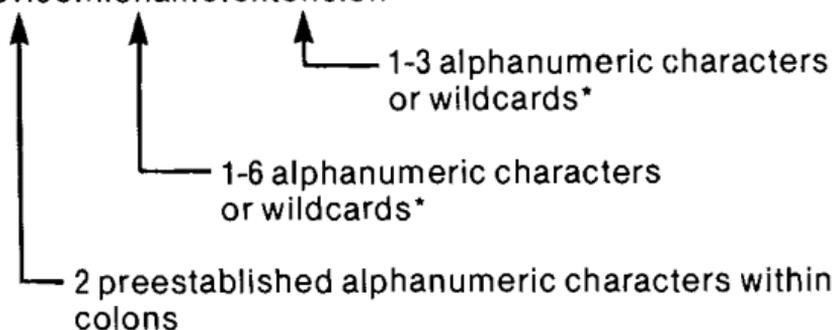
file=:device:

or

file=[:device:]filename[.extension]

DEVICE FILENAME FORMAT

:device:filename.extension



*wildcards:

An asterisk (*) matches any sequence of characters.

A question mark (?) matches any single character.

System Designated Device Names:

- :F0: thru :F9: Disk Drives
- :TI: Teletypewriter keyboard
- :TO: Teletypewriter printer
- :TP: Teletypewriter punch
- :TR: Teletypewriter reader
- :VI: Video terminal keyboard
- :VO: Video terminal screen
- :HP: High-speed paper tape punch
- :HR: High-speed paper tape reader
- :LP: Line printer
- :CI: Console input
- :CO: Console output
- :BB: Byte bucket

CONTROL CHARACTERS

- RUBOUT Deletes preceding character
- CTRL-E In a SUBMIT file, switches the console input from the command sequence file to the initial system console
- CTRL-P Allows literal entry of control characters (including itself)
- CTRL-Q Resumes console display
- CTRL-R Redisplay current input line as modified
- CTRL-S Stops console display
- CTRL-X Deletes all characters since last carriage return
- CTRL-Z Enters end-of-file

COMMAND SYNTAX

Disk Maintenance

FORMAT—format a new disk and copy files

FORMAT <device> <label> [<switches>]

where <device> = name of drive :F0:-:F9:

<label> = name of disk

<switches> = A—copy all files

S—copy files with system attribute set

FROM n—identifies disk containing files needed for formatting

IDISK—format a new disk as a basic system or non-system disk

IDISK <device> <label> [<switches>]

where <device> = name of drive :F0:-:F9:

<label> = name of disk

<switches> = S—copy files needed for basic system disk

P—specifies single drive mode

FROM n—identifies disk containing files needed for formatting

FIXMAP—map bad sectors on a hard disk

FIXMAP <drive>

where <drive> = number of hard disk unit 0-3

Subcommands are:

MARK <disk address> Change the known state of a sector from good to bad.

FREE <disk address>	Change the known state of a sector from bad to good.
LIST [<filename>]	List all known bad sectors.
COUNT	List the number of known bad sectors.
RECORD	Record changes specified by MARK and FREE.
QUIT	Exit to ISIS-II without recording changes.
EXIT	Record changes and exit to ISIS-II.

where <disk address> is given as:

<track> <sector> [T]
 <track> = 0-199 <sector> = 1-144
 T = process 36 sectors at once

File Maintenance

ATTRIB—change and/or display the attributes of a disk file

ATTRIB <file> [<attriblist>] [Q]

where <attriblist> is: I0 or I1—invisible
 W0 or W1—write protect
 F0 or F1—format
 S0 or S1—system

COPY—copy a file from one device to another

COPY <infile1> [, <infile2>, ..., <infilen>] TO
 <outfile> [<switches>]

where <switches> are:

- U = update—existing <outfile> automatically overwritten
- S = system—copy only files with S attribute
- N = nonsystem—copy only files without F or S attribute
- P = pause—single drive mode
- Q = query—query before each copy
- C = attribute—<outfile> created with same attributes as <infile>
- B = brief—existing <outfile> automatically deleted

HDCOPY—copy the contents of one hard disk to another

HDCOPY <drive1> TO <drive2> | BACKUP. <drive1> specifies the source disk. <drive2> specifies the destination disk. The BACKUP option can be used to backup a removable hard disk.

DELETE—remove references to a file from the directory

DELETE <filename 1>[Q][, <filename 2>[Q]][, <filename 3>[Q]][, ...][P]

where Q = query—query before each deletion
 P = pause—single drive mode

DIR—output the names of and information about the files listed within the disk directory

DIR [FOR <file>][TO <listfile>] [<switch>]

where <switch> is:

- 0-9 —indicates drive number
- I —invisible—list invisible files
- F —fast—list only name.ext of files
- P —pause—single drive mode
- O —single column display
- Z —show number of sectors in use

RENAME—change the name of a disk file

RENAME <oldname*> TO <newname*>

*Note—device must be the same in oldname and newname.

VERS—display ISIS utility version numbers

VERS [:Fn:]<filename>

where <filename> is the name of the ISIS file on :Fn: whose version number is to be displayed.

Code Conversion

HEXOBJ—convert a program from hexadecimal to ISIS-II format

HEXOBJ <hexfile> TO <absfile> [START (addr)]

OBJHEX—convert a program from ISIS-II to hexadecimal format

OBJHEX <absfile> TO <hexfile>

Program Execution

filename—execute the named program

<programe> [<parameters>]

where <parameters> are parameters needed by <programe>

DEBUG—load a program and give control to the monitor

DEBUG [<programe> [<parameters>]]

where <programe> is the name of the absolute object module to be debugged.

<parameters> are parameters needed by <programe>

SUBMIT—enter a file that contains commands to be executed

SUBMIT <name> [.<extension>] [(<parameter 0>, <parameter 1>, ..., <parameter 9>)]

where <name> [.extension] is the name of the file containing the command sequence definition. If .extension is omitted, **SUBMIT** looks for <name>.CSD.

<parameter n> specifies real values that replace formal parameters in the command sequence definition.

Program Control

LIB—create and control program libraries

LIB

CREATE <file>

ADD <filename>[(<modname>, ...)][, ...] **TO** <libfile>

DELETE <libfile>(<modname>, ...)

LIST <libfile>[(<modname>, ...)][, ...] [**TO**<listfile>]

[**PUBLICS**]

EXIT

LINK—combine program files and resolve external addressing

LINK <inputlist> **TO** <outputfile> [<controls>]

where <inputlist> can be:

<filename>[(<modname1>, <modname2>, ..., <modnamen>)]

or

PUBLICS (<filename>, <filename2>, ..., <filenamen>)

Controls: **MAP**

NAME (modname)

PRINT (filename)

LOCATE—convert relocatable object to absolute addresses for execution

LOCATE <inputfile> [**TO**<outputfile>] [<controls>]

where <controls> are:

MAP

DATA (address)

COLUMNS (number)

STACK (address)

PRINT (file)

MEMORY (address)

SYMBOLS

/common name/(address)

LINES

//(address)

PUBLICS

RESTART0

PURGE

START (address)

ORDER(segment sequence)

STACKSIZE (value)

CODE (address)

NAME (name)

File Editing

EDIT—create and modify ISIS-II files

EDIT <filename1>[TO<filename2>]

The TO <filename2> option causes the modified version of <filename1> to be stored in <filename2>.

TEXT POINTER:

B—beginning
Z—end
L—line
C—character
F—find

TEXT COMMANDS:

I —insert
S—substitute
D—delete
K—kill

TYPING A LINE:

T—type

TERMINATING AND SAVING A FILE:

E—exit
Q—quit
W—write

READING DATA:

A—Append

DETERMINE AVAILABLE MEMORY:

M—memory

ISIS-II SYSTEM CALLS

User must link object program with SYSTEM.LIB if making an ISIS System Call

ASSEMBLY LANGUAGE CALLS:

Interface accomplished by calling a single ISIS entry point (labeled ISIS) and passing two parameters:

Parameter 1: System Call Identifier (passed in register C).

SYSTEM CALL	IDENTIFIER	SYSTEM CALL	IDENTIFIER
OPEN	0	CONSOL	8
CLOSE	1	EXIT	9
DELETE	2	ATTRIB	10
READ	3	RESCAN	11
WRITE	4	ERROR	12
SEEK	5	WHOCON	13
LOAD	6	SPATH	14
RENAME	7		

Parameter 2: address of control block containing additional parameters for the call (passed in register pair DE).

NOTE:

addr = pass ADDRESS of parameter

val = pass VALUE of parameter

Every parameter must be passed as a two byte quantity.

ATTRIB—change the attributes of a disk file

file—addr

attribute—val (0 = invisible, 1 = system,
2 = write protect, 3 = format)

onoff—val (0 = reset, 1 = set)

status—addr

CLOSE—terminate input/output operations on a file

aftn—val

status—addr

CONSOL—change console device

infile—addr

outfile—addr

status—addr

DELETE—delete a file from the disk directory

file—addr

status—addr

ERROR—output error message on system console

errnum—val

(assembly language only) status—addr

EXIT—terminate program and return to ISIS-II

(assembly language only) status—addr

GETATT—obtain attribute information

path—addr

attribute—addr

status—addr

GETD—obtain file device directory

did—addr

conn—addr

count—addr

actual—addr

table—addr

status—addr

LOAD—load a file of executable code and transfer control
file—addr
bias—val
control—val (0 = calling program,
1 = loaded program, 2 = Monitor)
entry—addr
status—addr

OPEN—initialize file for input/output operations
aftnptr—addr
file—addr
access—val (1 = read, 2 = write, 3 = read and
write)
echo—val
status—addr

READ—transfer data from file to memory
aftn—val
buffer—addr
count—val
actual—addr
status—addr

RENAME—change disk filename
oldfile—addr
newfile—addr
status—addr

RESCAN—position marker to beginning of line
aftn—val
status—addr

SEEK—position disk file marker
aftn—val
mode—val (0 = return current position,
1 = move backward, 2 = move to specified position,
3 = move forward, 4 = move to EOF)
blockno—addr
byteno—addr
status—addr

SPATH—obtain file information
file—addr
buffer—addr
status—addr

WHOCON—determine file assigned as system console
aftn—val
buffer—addr
(assembly language only) status—addr

WRITE—transfer data from memory to file

aftn—val

buffer—addr

count—val

status—addr

ISIS-II ERROR MESSAGES

- 0 No error detected.
- *1 Limit of 19 buffers exceeded.
- 2 AFTN does not specify an open file.
- 3 Attempt to open more than 6 files simultaneously
- 4 Illegal pathname specification.
- 5 Illegal or unrecognized device specification in pathname.
- 6 Attempt to write to a file open for input.
- *7 Operation aborted; insufficient disk space.
- 8 Attempt to read from a file open for output.
- 9 No more room in disk directory.
- 10 Pathnames do not specify the same disk.
- 11 Cannot rename file; name already in use.
- 12 Attempt to open a file already open.
- 13 No such file.
- 14 Attempt to open for writing or to delete or rename a write-protected file.
- *15 Attempt to load into ISIS-II area or buffer area.
- 16 Illegal format record.
- 17 Attempt to rename/delete a non-disk file.
- *18 Unrecognized system call.
- 19 Attempt to seek on a non-disk file.
- 20 Attempt to seek backward past beginning of a file.
- 21 Attempt to rescan a non-lined file.
- 22 Illegal ACCESS parameter to OPEN or access mode impossible for file specified.
- 23 No filename specified for a disk file.
- *24 Disk error (see below).
- 25 Incorrect specification of echo file to OPEN.
- 26 Incorrect second argument in ATTRIB system call.
- 27 Incorrect MODE parameter in SEEK system call.
- 28 Null file extension.
- *29 End of file on console input.
- *30 Drive not ready.
- 31 Attempted seek on write-only (output) file.
- 32 Can't delete an open file.
- *33 Illegal system call parameter.
- 34 Bad third parameter argument to LOAD system call.
- 35 Attempt to extend a file opened for input by seeking past end-of-file.

*fatal error

- 201 Unrecognized switch.
- 202 Unrecognized delimiter character.
- 203 Invalid command syntax.
- 204 Premature end-of-file.
- 206 Illegal disk label.
- 207 No END statement found in input.
- 208 Checksum Error.
- 209 Illegal records sequence in object module file.
- 210 Insufficient memory to complete job.
- 211 Object module record too long.
- 212 Bad object module record type.
- 213 Illegal fixup record specified in object module file.
- 214 Bad parameter in a SUBMIT file.
- 215 Argument too long in a SUBMIT invocation.
- 216 Too many parameters in a SUBMIT invocation.
- 217 Object module record too short.
- 218 Illegal object module record format.
- 219 Phase error in LINK.
- 220 No end-of-file record in object module file.
- 221 Segment overflow during Link operation.
- 222 Unrecognized record in object module file.
- 223 Fixup record pointer is incorrect.
- 224 Illegal records sequence in object module file in LINK.
- 225 Illegal module name specified.
- 226 Module name exceeds 31 characters.
- 227 Command syntax requires left parenthesis.
- 228 Command syntax requires right parenthesis.
- 229 Unrecognized control specified in command.
- 230 Duplicate symbol found.
- 231 File already exists.
- 232 Unrecognized command.
- 233 Command syntax requires a "TO" clause.
- 234 File name illegally duplicated in command.
- 235 File specified in command is not a library file.
- 236 More than 249 common segments in input files.
- 237 Specified common segment not found in object file.
- 238 Illegal stack content record in object file.
- 239 No module header in input object file.
- 240 Program exceeds 64K bytes.

When error number 24 occurs, an additional message is output to the console:

```
STATUS=00nn  
D=x T=yyy S=zzz
```

where:

```
x   is the drive number  
yyy is the track address  
zzz is the sector address
```

and where nn has the following meanings for floppy disks:

- 01 Deleted record.
- 02 Data field CRC error.
- 03 Invalid address mark.
- 04 Seek error.

- 08 Address error.
- 0A ID field CRC error.
- 0E No address mark.
- 0F Incorrect data address mark.
- 10 Data overrun or data underrun.
- 20 Attempt to write on write protected drive.
- 40 Drive has indicated a write error.
- 80 Drive not ready.

For hard disks, nn means:

- 01 ID field miscompare.
- 02 Data field CRC error.
- 04 Seek error.
- 08 Bad sector address.
- 0A ID field CRC error.
- 0B Protocol violations.
- 0C Bad track address.
- 0E No ID address mark or sector not found.
- 0F Bad data field address mark.
- 10 Format error.
- 20 Attempt to write on write protected drive.
- 40 Drive has indicated a write error.
- 80 Drive not ready.

HEXADECIMAL-DECIMAL CONVERSION

BYTE				BYTE			
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0	0	0
1	4,096	1	256	1	16	1	1
2	8,192	2	512	2	32	2	2
3	12,288	3	768	3	48	3	3
4	16,384	4	1,024	4	64	4	4
5	20,480	5	1,280	5	80	5	5
6	24,576	6	1,536	6	96	6	6
7	28,672	7	1,792	7	112	7	7
8	32,768	8	2,048	8	128	8	8
9	36,864	9	2,304	9	144	9	9
A	40,960	A	2,560	A	160	A	10
B	45,056	B	2,816	B	176	B	11
C	49,152	C	3,072	C	192	C	12
D	53,248	D	3,328	D	208	D	13
E	57,344	E	3,584	E	224	E	14
F	61,440	F	3,840	F	240	F	15

MONITOR COMMANDS

Monitor I/O Configuration Command

A—Assign Command

A <logical device> = <physical device>

Possible values of logical and physical device are:

Logical Device	Physical Device
C or Console	T or TTY C or CRT B or BATCH 1 (reserved)
R or Reader	T or TTY P or PTR 1 or 2 (reserved)
P or Punch	T or TTY P or PTP 1 or 2 (reserved)
L or List	T or TTY C or CRT L or LPT 1 (reserved)

Q—Query Command

Q

Memory Control Commands

D—Display Memory

D <low address>, <high address>

F—Fill Memory

F <low address>, <high address>, <constant>

M—Move Memory

M <start addr>, <end addr>, <destination addr>

S—Substitute Memory

S <address>, [<data-byte>][,<data-byte>]][...]

Register Commands

X—Register Command

Display Form: X

Modify Form:

X <register>,<data>][,<data>]][,...]

Paper Tape I/O Commands

R—Read

R <bias>

W—Write

W <start address>, <end address>

E—End of File

E <entry point>

N—Null

N

Execute Command

G—Execute Command

G [<start address>][,<breakpoint 1>[,<breakpoint 2>]]

Utility Command

H—Hexadecimal add and subtract

H <number 1>, <number 2>

Monitor I/O Interface Routines

User must link object program with SYSTEM.LIB if using the following I/O routines.

Input Routines:

return character input as byte value in PL/M
or in the A Register in ASM

Output Routines:

pass character output as a byte parameter in PL/M
or in the C Register in ASM

CI—Console Input Routine

CO—Console Output Routine

LO—List Output Routine

RI—Reader Input Routine

PO—Punch Output Routine

UI—Universal PROM Programmer Input Routine

UO—Universal PROM Programmer Output Routine

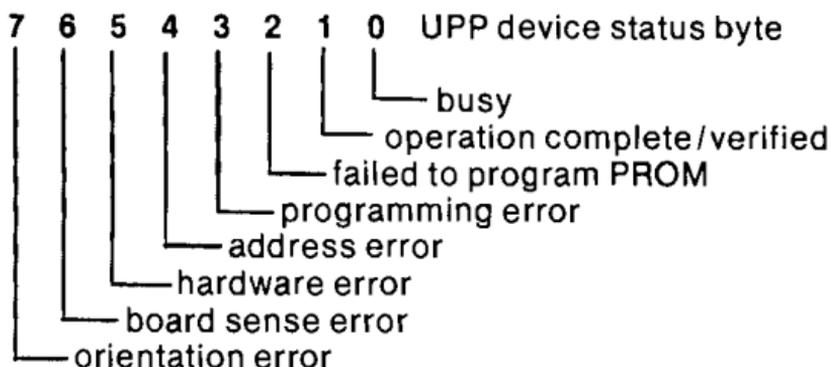
System Status Routines

User must link object program with SYSTEM.LIB if using the following status routines.

CSTS—Console Input Status Routine

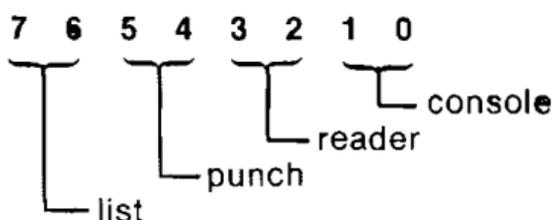
00H—no key pressed. 0FFH—key pressed.
returned as byte value in PL/M
or in the A register in ASM

UPPS—Universal PROM Programmer Status Routine



returned as byte value in PL/M
or in the A Register in ASM

IOCHK—Check System I/O Configuration Routine
 Physical Device Assigned to Logical System Device
 returned as byte value in PL/M
 or in the A register in ASM



VALUE	CONSOLE	READER	PUNCH	LIST
00	TTY	TTY	TTY	TTY
01	CRT	H.S. READER	H.S. READER	CRT
10	BATCH	Reserved	Reserved	LINE PRINTER
11	Reserved	Reserved	Reserved	Reserved

IOSET—Set System I/O Configuration Routine
 passed as byte value in PL/M
 or in the C register in ASM

MEMCK—Check RAM Size Routine
 value returned as address value in PL/M
 or in the H and L registers in ASM



3065 Bowers Avenue, Santa Clara, California 95051
 (408) 987-8080

Printed in U.S.A.

A460/1181/15K SVP