



# **iPDS™ INTEL PERSONAL DEVELOPMENT SYSTEM POCKET REFERENCE**

---



<b>Contents</b>	<b>Page</b>
Conventions .....	1
Entry Editing .....	1
Other Operating System Controls .....	1
Dual Processing on the iPDS System .....	2
Terms .....	2
Query Option .....	3
Pause Option .....	3
Attributes .....	3
Wildcard Filenames .....	3
ISIS-PDS Console Commands .....	3
CREDIT Text Editor .....	9
Changing Modes .....	9
General Editing Features .....	9
Screen Editing Commands .....	10
Command Line Editing Commands .....	10
Macro Editing Commands .....	12
Command Iteration .....	12
Conditional Editing Commands .....	12
Disk File Editing Commands .....	13
Alter Editing Commands .....	14
DEBUG Software Development Debugging Aid .....	15
Special DEBUG Syntax Terms .....	16
DEBUG Logical and Physical Devices .....	16
DEBUG Command Summary .....	17
Program Development Commands .....	20
System Programming Information .....	21
Summary of System Calls .....	23
Non-fatal Error Numbers .....	25
Fatal Error Numbers .....	25
iPPS PROM Programming .....	25
iPPS Devices .....	26
Special iPPS Syntax Terms .....	26
iPPS Command Switches .....	27
iPPS Command Defaults .....	27
iPPS Console Commands .....	28
Hexadecimal to Decimal Conversion .....	31
Base Conversions .....	32
ASCII Code List .....	35
ASCII Code Definition .....	36
ASCII Code in Binary .....	37
Control Codes .....	37
Function Codes .....	38
Escape Sequences .....	38
Graphics Codes .....	38
Powers of Two and Sixteen .....	40

## Conventions

UPPERCASE	Must be entered exactly as shown.
<i>italics</i>	Variable information.
[     ]	Optional item.
...	Repeatable item.
{     }	One and only one entry must be selected.
<u>UNDERLINE</u>	Shortest valid abbreviation for keyword.

## Entry Editing

RUBOUT. Deletes the previous character entered.

CTRL-P. Allows literal entry of control characters.

CTRL-R. Redisplays current input line as modified.

CTRL-X. Deletes contents of line-editing buffer.

CTRL-Z. Enters end-of-file.

RETURN (Carriage Return). Terminates command input and submits line-editing buffer as a command.

ESC. Re-edit previous or present command line.

An+. Prompt in line editing mode

Bn+. Prompt in line editing mode

## Other Operating System Controls

CTRL-E. Switches console input from a SUBMIT file.

CTRL-Q. Resumes console display.

CTRL-S. Stops console display.

CTRL-B. Alternate Escape character.

FUNC-S. Slow down CRT scroll.

FUNC-T. Switch to typewriter mode.

FUNC-R. Software reset processor.

FUNC-0 thru FUNC-9. User defined JOB command.

## Dual Processing on the iPDS System

An > . Prompt for Processor A.

Bn > . Prompt for Processor B.

FUNC-HOME. Switches the keyboard between processors.

FUNC-↑. Increases by one line the display size of the bottom half of the CRT screen.

FUNC-↓. Decreases by one line the display size of the bottom half of the CRT screen.

### Terms

*logical device.* :nn: where nn can be:  
:F0: thru :F5: Logical Disk Devices  
:CI: Console Input  
:CO: Console Output  
:BB: Byte Bucket

*physical device.* 1 thru 3 Physical Disk Drives  
4 and 5 Bubble Memory Physical Device  
Number  
:VI: Video Terminal Keyboard  
:VO: Video Terminal CRT  
:SI: Serial Input Device  
:SO: Serial Output Device  
:TR: Teletype Paper Tape Reader  
:TP: Teletype Paper Tape Punch  
:HR: High Speed Paper Tape Reader  
:HP: High Speed Paper Tape Punch  
:LP: Line Printer  
:L1: User Defined List Device  
:I1: User Defined Console Input  
:O1: User Defined Console Output  
:R1: User reader input device 1  
:P1: User punch output device 1  
:R2: User reader input device 2  
:P2: User punch output device 2

*filename.* name.extension where name is 1 to 6 characters and extension is 1 to 3 characters.

*pathname.* :device name:filename; abbreviated pn

*source.* input file

*destination.* output file

*jobfile.* file containing operating system commands

*multimodule row.* 0 for row J1/J2 or 1 for row J3/J4

*valid.* 1 to 6 character name of disk volume

*volext.* 1 to 3 character extension for disk volume

## Query Option

Q. For many commands a Q on the command line prompts the user before processing the file.

## Pause Option

P. For many commands a P on the command line allows the command to be used on single drive systems.

## Attributes

The file attributes are:

W	Write Protect
S	System
F	Format
I	Invisible
J	User Defined
K	User Defined
L	User Defined

## Wildcard Filenames

\*. In a filename specifies a match to any characters and any number of characters in that position.

?. In a filename specifies a match to any single character in that position except blanks.

## ISIS-PDS Console Commands

**ASSIGN.** Map logical devices to physical devices.

```
ASSIGN  { :CI:  TO  cid }
         { :CO:  TO  cod }
         { ldd  TO  pdd }
```

*cid.* :VI:, :SI:, :I1:, :BB:, or pathname of a disk file.

*cod.* :VO:, :SO:, :O1:, :BB:, or pathname of a disk file.

*ldd.* :F0:, :F1:, :F2:, :F3:, :F4:, or :F5:.

*pdd.* 0, 1, 2, 3, 4, or 5.

Examples:

```
ASSIGN :F1: TO 4
ASSIGN :CO: TO :SO:
```

## ATTACH. Assign multimodule row to a processor.

ATTACH *multimodule row*

Example:

ATTACH 1

## ATTRIB. Display and modify attributes of disk files.

ATTRIB *pathname*  $\left[ \begin{array}{c} Wn \\ In \\ Sn \\ Fn \\ Jn \\ Kn \\ Ln \end{array} \right] \dots [Q]$

- n.* 0 for reset and 1 for set.
- W. Write Protect.
- S. System.
- F. Format.
- I. Invisible.
- J, K, L. User Defined.
- Q. Prompt before processing.

Example:

ATTRIB :F1:MYFILE.TXT W0 I0 S0 F0 J1 K1 L1

## COPY. Transfer files.

COPY *source pn* TO *destination pn*  $\left[ \begin{array}{c} \{S\} \\ \{N\} \end{array} \right] \left[ \begin{array}{c} \{B\} \\ \{U\} \end{array} \right] \left[ \begin{array}{c} \{C\} \\ \{P\} \\ \{Q\} \end{array} \right] \dots \left[ \begin{array}{c} \{J\} \\ \{K\} \\ \{L\} \end{array} \right] \dots$

- J. Copy only files with User Defined attribute J.
- K. Copy only files with User Defined attribute K.
- L. Copy only files with User Defined attribute L.
- S. Copy only files with System attribute.
- N. Copy only files without S or F attribute.
- B. No prompt if destination exists. Delete existing file; copy source to newly created destination.
- U. Same as B except existing file is not deleted first.
- C. Copy the source file's attributes.
- P. Single drive COPY.
- Q. Prompt before processing.

Examples:

COPY MYFILE.TXT TO :F2:OLD.TXT  
COPY :F3:\*.\* TO :F2: NCQ

### **COPY. Append files.**

COPY *source pn 1,source pn 2[,source pn 3, . . .,source pn n]*

TO *destination pn*  $\left[ \begin{array}{c} \{B\} \\ \{U\} \end{array} \right]$  [P]

B. No prompt if destination exists. Delete existing file; append source to newly created destination.

U. Same as B except existing file is not deleted first.

P. Single drive append.

Example:

COPY :F2:FILE1;:F4:FILE2;:F3:FILE3 TO :F0:FILE 4

### **DELETE. Remove files from the disk.**

DELETE *pathname 1[,pathname 2, . . . , pathname n]*  $\left[ \begin{array}{c} \{Q\} \\ \{P\} \end{array} \right]$

Example:

DELETE :F4:PROG?.SRC Q

### **DETACH. Release multimodule row from processor.**

DETACH *multimodule row*

Example:

DETACH 1

### **DIR. Display index of disk files.**

DIR [TO *pn*] [FOR *pn*] [*n*] [I]  $\left[ \begin{array}{c} \{J\} \\ \{K\} \\ \{L\} \end{array} \right] \dots \left[ \begin{array}{c} \{F\} \\ \{O\} \\ \{P\} \\ \{Z\} \end{array} \right] \dots$

TO *pn*. Device to receive directory listing.

FOR *pn*. Scope of the directory listing.

*n*. Logical device from which files are listed.

I. Include Invisible files.

J. Only files with User Defined attribute J included.

K. Only files with User Defined attribute K included.

L. Only files with User Defined attribute L included.

F. Fast listing; only filenames and extensions.

O. Single column listing.

P. Single drive directory.

Z. Only summary line is listed.

Examples:

DIR

DIR 1 I

DIR TO :LP: FOR :F0:PROG?.SRC

**ENDJOB. Terminate a file used as console input.**

ENDJOB *[comment]*

**Examples:**

ENDJOB JOB9  
ENDJOB

**HELP. Display help information for ISIS-PDS commands.**

HELP  $\left[ \begin{array}{l} \{n\} \\ \{topic\} \end{array} \right]$

*n*. Number of an ISIS error message.

*topic*. Term on main HELP screen for which further help is available.

**Examples:**

HELP COPY  
HELP 29  
HELP

**IDISK. Initialize a disk or bubble memory for ISIS-PDS use.**

IDISK :Fn:volid[.volext]  $\left[ \begin{array}{l} \{S\} \\ \{P\} \end{array} \dots \right]$

S. System disk.

**Examples:**

IDISK :F2:VOL2 S

**JOB. Batch commands and execute from a file.**

JOB *pathname*

RETURN. As the first character after the = prompt, all commands entered are written to *filename*. An ENDJOB command is appended to the last command. Commands are executed from *filename*.

CTRL-Z. Returns to standard input mode and deletes any commands already entered.

ESC. As the first character after the = prompt, causes the commands to be saved in *filename*, but does not execute from the file.

**Example:**

JOB :F1:JOB1



**RENAME. Change the filename or extension of a disk file.**

RENAME *old pathname* TO *new pathname*

Example:

RENAME PROGA.SRC TO PROGB.SRC

**SERIAL. Configure the 8251 USART for the serial output port.**

For synchronous mode, the format is:

SERIAL S [P = *a* W = *n1* E = *n2* C = *n3* I = *n4*]

For asynchronous mode, the format is:

SERIAL A [P = *a* W = *n1* S = *n2* B = *n3* I = *n4*]

**S. Synchronous mode of data transfer.**

**P = *a*.** Parity. The value of *a* can be E for even parity, O for odd parity, N for no parity.

**W = *n1*.** Word size. The value of *n1* is the word size in bits: 5, 6, 7, or 8.

**E = *n2*.** External synchronization. The value of *n2* can be 0 for output or 1 for input.

**C = *n3*.** Number of synchronization characters. The value of *n3* can be 0 for double synchronization character or 1 for single sync character.

**I = *n4*.** Instruction for the 8251 USART. The value of *n4* is the code for the 8251 instruction.

**A. Asynchronous mode of data transfer.**

**S = *n2*.** Number of stop bits. The value of *n2* can be 1 for one stop bit, 1.5 for one and a half stop bits, 2 for two stop bits.

**B = *n3*.** Baud rate. The value of *n3* can be 110, 150, 300, 600, 1200, 2400, 4800, 9600, or 19200.

Examples:

SERIAL S P = E W = 5 E = 0 C = 0  
SERIAL A P = 0 S = 2 W = 5 B = 300

**SUBMIT. Execute commands from a disk file.**

SUBMIT *pn* [(*parameter 0*, *parameter 1*, . . . , *parameter 9*)]

Example:

SUBMIT JOB99(PROGA)

**[?] . Display version of command line interpreter.**

?

Example:

?  
CLI V1.0

**[@] . Display the contents of a file on the screen.**

@*pathname* [4]

4. tabs expanded to 4 spaces.

After entering command, the following keyboard commands can be used:

**P.** Page mode (Display file 20 lines at a time).

**S.** Slow scroll mode (Display file continuously scrolling at a slow speed).

**F.** Fast scroll mode (Display file continuously scrolling at a fast speed).

**E.** Exit back to the operating system.

**L.** Line-by-line mode (Display file one line at a time).

**B.** Back up 1K bytes and continue.

**Z.** Print the last 1K bytes of the file.

**CTRL-S.** Alternately pause and continue display.

**a.** Any key continues display after halted.

Example:

@ MYDOC.TXT

**[/] . Assign a file as the console input device.**

/*pathname*

Examples:

/:F1:CMDFIL  
/:S1:

**[#] . Re-assign console output to the CRT screen.**

#

**[.] . SUBMIT single command line from file with parameters.**

.pn [*parameter 0, parameter 1, . . . ., parameter 9*]

Example:

.:F1:CMDFIL

**FUNC** **n**. Assign the file **JOBn.CSD** as the console input device.

FUNC-*n*

*n*. from 0 to 9.

Example:

FUNC 0

**ESC**. Re-edit and re-execute previous or current command line.

ESC

After entering command, the following keyboard commands can be used:

ESC. Execute entire line.

RETURN. Execute line up to current cursor position.

CTRL-X. Terminate re-edit and return to ISIS.

← . Move cursor left.

→ . Move cursor right.

CTRL-B. Move cursor to beginning of line.

CTRL-A. Encloses characters to be inserted.

CTRL-D. Delete character at current cursor.

CTRL-L. Move cursor to end of line.

RUBOUT. Same as CTRL-D.

## CREDIT Text Editor

CREDIT *pathname1* [TO *pathname2*]  $\left[ \begin{array}{l} \{ \text{MACRO}[(\text{command file})] \} \\ \text{NOMACRO} \end{array} \right]$

Examples:

CREDIT MOD2.PLM NOMACRO  
CREDIT MOD2.PLM TO MOD3.PLM MACRO

## Changing Modes

TO SWITCH MODES. Press HOME key to enter command mode; press CTRL-V to enter screen mode.

## General Editing Features

TO MOVE CURSOR. Use the four arrow keys on keyboard.

TO MODIFY TEXT. Type new character over old character.

TO ENTER TEXT. Move the cursor to the end of file mark (vertical bar) or to a line terminator (up arrow). Type characters on the keyboard.

## Screen Editing Commands

**CTRL-A. Add text.**

CTRL-A*text*CTRL-A

**CTRL-C. Add character.**

CTRL-C*x*

**CTRL-Z. Delete text.**

CTRL-Z*move cursor*CTRL-Z

**CTRL-D. Delete character.**

CTRL-D

**CTRL-V. View page.**

CTRL-V

**CTRL-N. Next page.**

CTRL-N

**CTRL-P. Previous page.**

CTRL-P

## Command Line Editing Commands

**DC. Delete character command.**

DC { *number* }  
{ *tag* }

**DL. Delete line command.**

DL { *number* }  
{ *tag* }

**EQ. Quit command.**

EQ

**EX. Exit command.**

EX [*filename*]

**F. Find command.**

*Fdelimiter string delimiter*     $\left. \begin{array}{l} \{ number \\ \} \\ \{ tag \\ \} \end{array} \right\}$

**H. Help command.**

H

**I. Insert command.**

*Idelimiter text delimiter*

**J. Jump character command.**

J     $\left. \begin{array}{l} \{ number \\ \} \\ \{ tag \\ \} \end{array} \right\}$

**L. Line command.**

L[*number*]

**P. Print command.**

P     $\left. \begin{array}{l} \{ number \\ \} \\ \{ tag \\ \} \end{array} \right\}$

**PH. Print hexadecimal command.**

PH     $\left. \begin{array}{l} \{ number \\ \} \\ \{ tag \\ \} \end{array} \right\}$

**S or SQ. Substitute command.**

S[Q]*delimiter oldtext delimiter newtext delimiter*     $\left. \begin{array}{l} \{ number \\ \} \\ \{ tag \\ \} \end{array} \right\}$

**TS. Tag set command.**

TS*n*

**TD. Tag delete command.**

TD*n*

**XM. Move command.**

XM *tag1,*     $\left. \begin{array}{l} \{ number \\ \} \\ \{ tag2 \\ \} \end{array} \right\}$

**XC. Copy command.**

XC *tag1,*     $\left. \begin{array}{l} \{ number \\ \} \\ \{ tag2 \\ \} \end{array} \right\}$

## Macro Editing Commands

### MD. Macro delete command.

MD    { *name* }  
          \*

where \* is used to delete all currently defined macros. If *name* is specified, asterisk cannot be used.

### MF or CTRL-F. Macro function command.

{ MF*name*[(*parameter 1* [, *parameter 2* , . . . , *parameter n*])] }  
  CTRL-F*name*  
  *name*

### MS. Macro set command.

MS*name delimiter text delimiter*

### ?M. Display macro command.

?M

## Command Iteration

{ <sup>!</sup>*number* } *delimiter 1 commands delimiter 2*

where *commands* are executed *number* of times or forever (!).

## Conditional Editing Commands

### EL. Exit loop command.

EL

### QU. Set query flag command.

QU

### QT and QF. Conditional execution of commands based on query flag.

{ QT;*command* }  
  { QF;*command* }

### U. Write user message command.

U*delimiter text delimiter*

**YT and YF. Conditional execution commands based on yes flag.**

$\{YT;command\}$   
 $\{YF;command\}$

## **Disk File Editing Commands**

**B. Begin file command (move file pointer to BOF).**

B

**CR or CW. Close file commands.**

CR  
CW

**G. Get command (execute disk file with edit commands).**

G *pathname*

**OR. Open disk file for read command.**

OR *pathname*

**OW. Open disk file for write command.**

OW *pathname*

**R. Read file command.**

R[*number*]

**W. Write file command.**

W  $\left[ \begin{array}{l} \{number\} \\ \{tag\} \end{array} \right]$

## Alter Editing Commands

### A. Alter environment command.

*Acode = value*

The *value* should be given as a single character except for V and T which are given as decimal numbers and B, O, and V which are given as a hexadecimal number. The possible *codes* are:

- L. Assign line terminator character. Default: 5EH (up arrow).
- V. Number of lines to display on each screen (22, 23, 24, 25). Default: 24 lines.
- S. Suppress NOT FOUND message (T) or display it (F). Default: F (not suppressed).
- T. Assign the number of spaces for a TAB (0-79). Default: 8.
- B. Assign character to use as BREAK character. Default: 1BH (ESC).
- C. Assign character to display for non-printing characters in screen mode. Default: 5EH (up arrow).
- W. Specify status of line wrap feature. Default: T (available).
- O. Offset used if direct cursor addressing is in effect. Default: 20H.
- Q. Literalizing character. Default: 5CH (backslash).
- X. Specify whether row-major or column major order is in effect if direct cursor addressing is used. Default: F (row major).

### AF. Alter function command.

*AFcode = value*

The *value* is given as a single character for WA, WC, WJ, and BK; the other codes take a zero to four byte hexadecimal value. The *codes* are:

- WA. Wildcard character for matching any number of characters. Default: CTRL-Y.
- WC. Wildcard character for matching either upper or lower case characters. Default: CTRL-W.
- WJ. Wildcard character for matching any single character. Default: ?.
- BK. Code which blanks out a single screen location. Default: Space.
- AC. Specifies codes to use as command to directly address cursor. Default: 00H.
- IG. Specifies byte to ignore when received from keyboard. Default: 00H.
- IL. No effect. For compatibility with other editors. Default: none.
- DL. No effect. For compatibility with other editors. Default: none.



**CD.** Assign cursor down code. Default: 1CH.  
**CH.** Assign cursor home code. Default: 1DH.  
**CL.** Assign cursor left code. Default: 1FH.  
**CR.** Assign cursor right code. Default: 14H.  
**CU.** Assign cursor up code. Default: 1EH.  
**MB.** Assign code to move cursor to beginning of line. Default: 0DH.  
**MD.** Assign cursor down code. Default: 1BH 42H.  
**MH.** Assign code to move cursor to home position. Default: 1BH 48H.  
**ML.** Assign code to move cursor left. Default: 1BH 44H.  
**MR.** Assign code to move cursor right. Default: 1BH 43H.  
**MU.** Assign code to move cursor up. Default: 1BH 41H.  
**EK.** Assign code to erase entire line. Default: 1BH 4BH.  
**EL.** Assign code to erase line following cursor. Default: 20H.  
**ER.** Assign code to erase screen following cursor. Default: 1BH 4AH.  
**ES.** Assign code to erase entire screen. Default: 1BH 45H.  
**XA.** Assign CTRL-A code. Default: 01H (CTRL-A).  
**XC.** Assign CTRL-C code. Default: 03H (CTRL-C).  
**XD.** Assign CTRL-D code. Default: 04H (CTRL-D).  
**XF.** Assign CTRL-F code. Default: 06H (CTRL-F).  
**XN.** Assign CTRL-N code. Default: 0EH (CTRL-N).  
**XP.** Assign CTRL-P code. Default: 10H (CTRL-P).  
**XV.** Assign CTRL-V code. Default: 16H (CTRL-V).  
**XX.** No effect. For compatibility with other editors. Default: none.  
**XZ.** Assign CTRL-Z code. Default: 1AH (CTRL-Z).

**?A. Display alter command.**

? A

## **DEBUG. Software development debugging aid.**

DEBUG [*command line*]

Examples:

```
DEBUG LIST FILE.TXT  
DEBUG
```

## Special DEBUG Syntax Terms

*logical device.* See tables below.

*physical device.* See tables below.

*start address.* Beginning address.

*end address.* Final address.

*constant.* Any Hexadecimal value.

*breakpoint.* Address at which program halts.

*destination address.* Output address.

*number.* Any Hexadecimal value.

*databyte.* Any Hexadecimal value.

*register.* See table below.

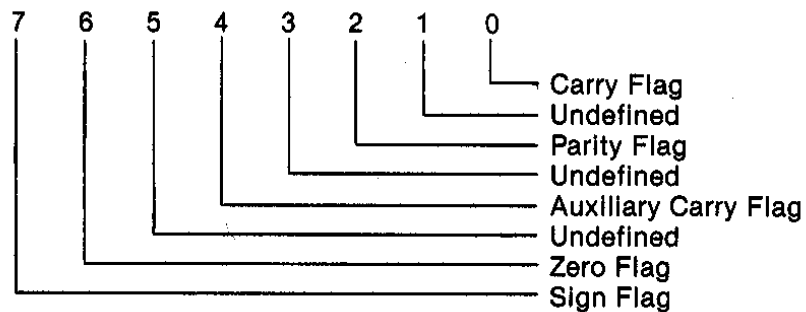
## DEBUG Logical and Physical Devices

Single Letter Symbol for logical device	Device
C	Console
R	Reader
P	Punch
L	List

Logical Device	Single Letter Symbol for physical device	Device
CONSOLE	S	Serial I/O Device
	C	CRT Terminal
	B	Batch Mode Device
READER	1	User Defined Device 1
	S	Serial I/O Device
	C	CRT Terminal
PUNCH	1	User Defined Device 1
	2	User Defined Device 2
	S	Serial I/O Device
LIST	C	CRT Terminal
	1	User Defined Device 1
	2	User Defined Device 2
	S	Serial I/O Device
	C	CRT Terminal
	L	Line Printer
	1	User Defined Device 1

Symbol	Register	Size of Register
A	CPU A Register	1 Byte
B	CPU B Register	1 Byte
C	CPU C Register	1 Byte
D	CPU D Register	1 Byte
E	CPU E Register	1 Byte
F	CPU Flag Byte	1 Byte
H	CPU H Register	1 Byte
L	CPU L Register	1 Byte
M	CPU H and L Registers Combined	2 Bytes
P	CPU Program Counter	2 Bytes
S	CPU Stack Pointer	2 Bytes

The Flag Byte is displayed in the following 8-bit format:



## DEBUG Command Summary

### A. Assign logical device to physical device.

*Logical device = physical device*

Example:

.AC = S

### C. Disassemble specified memory as instruction mnemonics.

*Cstart address,number*

Example:

.C8000,500

### D. Display specified memory range.

*Dstart address,end address*

Example:

.D0100,1000

### **E. Exit to ISIS-PDS.**

E

### **F. Fill memory with constant.**

*Fstart address,end address,constant*

Example:

.F4000,402F,00

### **G. Execute program with breakpoints.**

*G[start address] [,breakpoint 1[,breakpoint 2]]*

Examples:

.G  
.G4000,-40CF,-5000

### **H. Hexadecimal add and subtract.**

*Hnumber 1,number 2*

Example:

.HE49,111  
0F5A 0D38

### **I. Input byte from iPDS port.**

*Iport address*

Example:

.I0C0  
C0 => 0D

### **M. Move block of memory.**

*Mstart address,end address,destination address*

Example:

.M4000,4100,5000

## N. Execute a specified number of instructions.

*Nstep count*

Example:

```
.N4
400D B8      CMP B
400E C20640 JNZ 4006
4011 0E09    MVI C,09
4013 110040 LXI D,4000
```

## O. Output byte to I/O port.

*Oport address,databyte*

Example:

```
.O91,37
37 = > 91
```

## Q. Query current devices assigned.

Q

## S. Substitute memory interactively.

*Saddress,[databyte],[databyte] . . .*

Example:

```
.S4000,3A-,56-00,49-
```

## T. Disassemble memory relative to Program Counter.

*Tnumber*

Example:

```
.T25
```

## X. Display/modify registers.

**Display Form**

X

**Modify Form**

*Xregister,[data],[data] . . .*

Example:

```
.X
.XA,50
```

## Program Development Commands

These commands are described in the *MCS-80/85 Utilities User's Guide*, order number 121617.

### LINK. Combine relocatable modules.

```
LINK inputlist TO pathname [controls]  
inputlist. pathname [(module name, . . .)], . . . or PUBLICS  
(pathname[(module name, . . .)], . . .)  
controls. MAP, NAME(modulename), or PRINT(pathname)
```

Example:

```
LINK A, B, OBJECT.LIB TO C
```

### LOCATE. Assign absolute addresses to program modules.

```
LOCATE pathname [TO pathname] [controls]
```

*controls*. can be any of the following:

COLUMNS ( <i>number</i> )	START( <i>address</i> )
LINES	CODE( <i>address</i> )
MAP	DATA( <i>address</i> )
NAME( <i>module name</i> )	MEMORY( <i>address</i> )
PRINT( <i>pathname</i> )	STACK( <i>address</i> )
PUBLICS	STACKSIZE( <i>length</i> )
PURGE	/common/( <i>address</i> )
SYMBOLS	//( <i>address</i> )
RESTART0	ORDER( <i>segment sequence</i> )

Example:

```
LOCATE SAM.PLE TO EXAM.PLE MAP PRINT(TOPO.MAP)  
COLUMNS(3) PURGE
```

### LIB. Manage libraries containing program modules.

```
LIB
```

Subcommands are CREATE, ADD, DELETE, LIST, and EXIT.

Example:

```
LIB
```

### HEXOBJ. Convert hex format files to absolute object format.

```
HEXOBJ hexfile TO absfile [START(address)]
```

Example:

```
HEXOBJ :F1:PFILE.HEX TO PFILE.OBJ START(3300H)
```

**OBJHEX. Convert absolute object format files to hex format.**

OBJHEX *absfile* TO *hexfile*

Example:

```
OBJHEX PFILE.OBJ TO :F1:PFILE.HEX
```

## System Programming Information

EQU Values			
System Call	Number	System Call	Number
OPEN	0	EXIT	9
CLOSE	1	ATTRIB	10
DELETE	2	RESCAN	11
READ	3	ERROR	12
WRITE	4	WHOCON	13
SEEK	5	SPATH	14
LOAD	6	ATTACH	15
RENAME	7	DETACH	16
CONSOL	8		

*access*. OPEN. 2-byte number telling how the file is to be used, i.e., read or write or both.

*actual\$ptr*. READ. 2-byte pointer to actual number of bytes successfully read.

*atrb*. ATTRIB. 2-byte number indicating which attribute to change.

*block\$ptr*. SEEK. 2-byte pointer to the block number.

*buf\$ptr*. READ, WHOCON, WRITE. 2-byte pointer to the area declared for reading from (or writing to) a file; for read and write, it should be at least COUNT bytes long or undefined results will occur.

*byte\$ptr*. SEEK. 2-byte pointer to the byte number.

*char*. CO, LO, PO. Byte value output to console, serial device, or printer.

*ci\$path\$ptr*. CONSOL. 2-byte pointer to ASCII string containing pathname of the console input device.

*config\$byte*. IOSET. Byte value used to assign I/O devices.

*conn*. CLOSE, READ, RESCAN, SEEK, WHOCON, WRITE. 2-byte connection number to a file or device.

*conn\$ptr*. OPEN. 2-byte pointer to the connection number.

*control\$sw*. LOAD. 2-byte value indicating where to transfer control after the load.

*co\$path\$ptr*. CONSOL. 2-byte pointer to ASCII string containing pathname of the console output device.

*count*. READ, WRITE. 2-byte value that specifies the number of bytes to read from or write to a file.

*echo*. OPEN. 2-byte connection number for the echo file when a line edited file is opened.

*entry\$ptr*. LOAD. 2-byte address of location to which the loaded program should return for LOAD.

*entry\$point*. IODEF. 2-byte address of entry point of user written I/O driver.

*errnum*. ERROR. 2-byte error number to output to console.

*function\$code*. IODEF. Byte value that identifies which I/O driver is being added.

*info\$ptr*. SPATH. 2-byte pointer to the pathname whose connection number is needed.

*load\$offset*. LOAD. 2-byte offset value added to load address causing program to load at adjusted address.

*mode*. SEEK. 2-byte value representing direction and type of seek operation.

*mmio\$row*. ATTACH, DETACH. 2-byte value specifying which multimodule row is being attached or detached.

*newpath\$ptr*. RENAME. 2-byte pointer to the new pathname of the file being renamed.

*oldpath\$ptr*. RENAME. 2-byte pointer to the old pathname of the file being renamed.

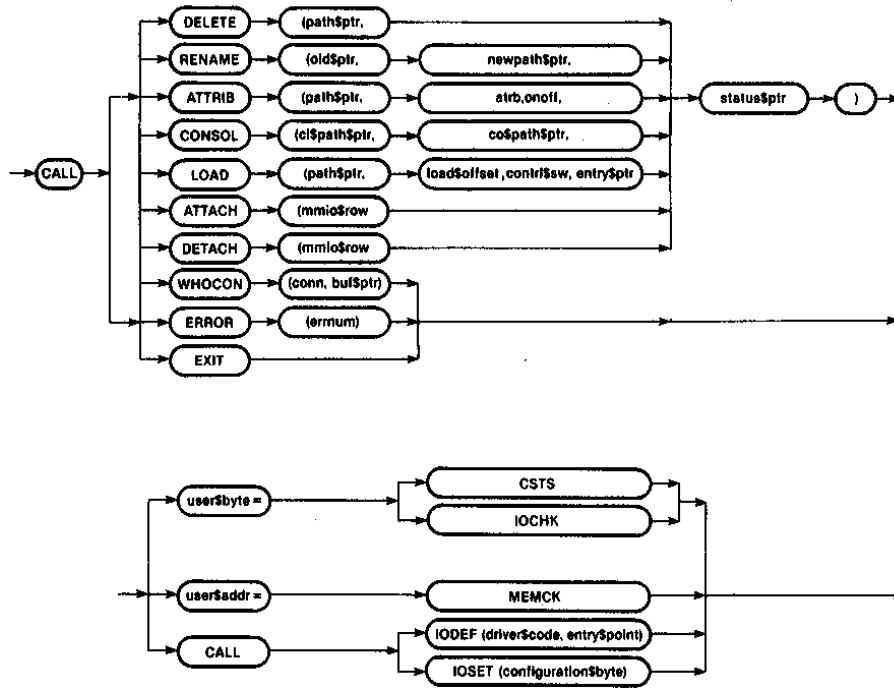
*onoff*. ATTRIB. 2-byte number indicating whether the attribute is to be set or reset.

*path\$ptr*. ATTRIB, DELETE, LOAD, OPEN, SPATH. 2-byte pointer to the pathname of the file being accessed.

*status\$ptr*. All routines but EXIT and the primitives. 2-byte pointer to the error numbers generated during the system call.

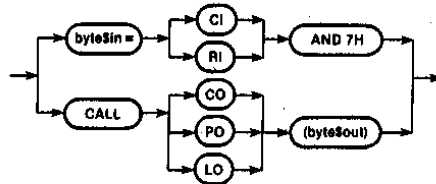
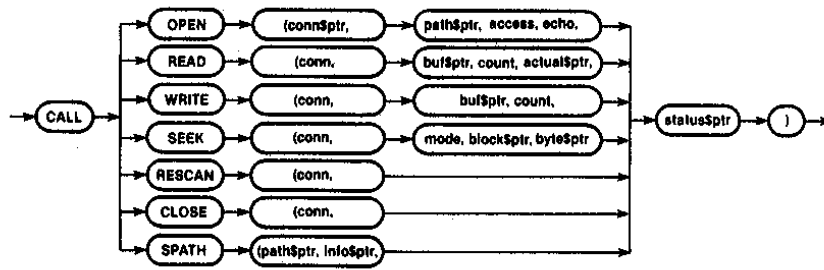


# Summary of System Calls



Procedure or Function	Name of Status Routine	Reports On	PL/M Results	Assembly Language Results
F	CSTS	Console Input Status	00H if no key, 0FFH if any key pressed	The values shown at left are returned in Reg. A
P	IODEF	Existence and location of user-written programs (I/O drivers) for non-standard devices	Nothing, IODEF tells the Monitor what routines are to be used.	
F	IOCHK	I/O Configuration	Eight-bit value describing the currently configured devices <sup>1</sup>	Value returned in the A-register
P	IOSET	New System Configuration	Eight-bit value is sent, setting new I/O assignments	Value is sent in the C-register
F	MEMCK	Highest address of contiguous memory available to user programs	The highest (address (a 16-bit value))	Address value returned in the H and L registers

<sup>1</sup>Bit meanings in routine description



Procedure or Function	Name of Monitor Routine	Performs	PL/M Results	Assembly Language Results
F	CI	Console Input	fills <i>byte\$p</i>	fills Reg. A with character from console input device
P	CO	Console Output	sends <i>byte\$out</i>	sends byte from Reg. C to console output device
F	RI	Reader Input	fills <i>byte\$p</i>	fills Reg. A with character from paper tape reader device
P	PO	Punch Output	sends <i>byte\$out</i>	sends byte from Reg. C to paper tape punch device
P	LO	List Output	sends <i>byte\$out</i>	sends byte from Reg. C to system list device

## Non-Fatal Error Numbers

OPEN	3, 4, 5, 9, 12, 13, 14, 22, 23, 25, 28, 63
READ	2, 8
WRITE	2, 6
SEEK	2, 19, 20, 27, 31, 35
RESCAN	2, 21
CLOSE	2
DELETE	4, 5, 13, 14, 17, 23, 28, 32
RENAME	4, 5, 10, 11, 13, 17, 23, 28
ATTRIB	4, 5, 13, 23, 26, 28
CONSOL	None; all errors are fatal
WHOCON	None
ERROR	None
LOAD	3, 4, 5, 12, 13, 22, 23, 28, 34
EXIT	None
SPATH	4, 5, 23, 28

## Fatal Error Numbers

OPEN	1, 7, 24, 30, 33
READ	24, 30, 33
WRITE	7, 24, 30, 33
SEEK	7, 24, 30, 33
RESCAN	33
CLOSE	33
DELETE	1, 24, 30, 33
RENAME	1, 24, 30, 33
ATTRIB	1, 24, 30, 33
CONSOL	1, 4, 5, 12, 13, 14, 22, 23, 24, 28, 30, 33, 63
WHOCON	33
ERROR	33
LOAD	1, 15, 16, 24, 30, 33
EXIT	None
SPATH	33
ATTACH	33, 60, 61
DETACH	33, 60, 61

## IPPS. PROM programming

[:Fn:]IPPS

Example:

IPPS

## iPPS Devices

**PROM.** Logical section of Programmable Read Only Memory in PROM device(s) plugged into the Personality Module.

**BUFFER.** Section of development system memory (created and maintained by the iPPS software) providing a temporary area where data can be held and modified.

**FILE.** Logical device containing data stored in an ISIS-PDS file specified within iPPS commands by the normal ISIS-PDS file specification format *:device:filename.extension*.

## Special iPPS Syntax Terms

*startaddr.* Starting address of a section of data; values range from 0 to 16777215 (i.e., 0 to  $2^{24} - 1$ ).

*endaddr.* Ending address of a section of data; values range from 0 to 16777215 (i.e., 0 to  $2^{24} - 1$ ). This value can be optionally entered as a length value in the form *Llength*. See the description of *length*.

*destaddr.* Destination start address for a section of data that will be written; values range from 0 to 16777215 (i.e., 0 to  $2^{24} - 1$ ). This value can be optionally entered as an offset value in the form *0offset*. See the description of *offset*.

*length.* Length (in bytes) of a section of data; values range from 0 to 16777216 (i.e., 0 to  $2^{24}$ ).

*offset.* Value used to calculate the destination start address for a section of data to be written. Can be positive or negative. The address is calculated by adding the *offset* to the *startaddr* for the source device and must range from 0 to 16777215 (i.e., 0 to  $2^{24} - 1$ ).

*file.* Standard ISIS-PDS file specification of the form *:device:filename.extension*. The following *filenames* are invalid:

B	P
BU	PR
BUF	PRO
BUFF	PROM
BUFFE	
BUFFER	

*byteval.* Specifies a constant to be loaded into each byte of the selected address range.

*switch.* Base Switch, Data Switch, File Switch, or Patch Switch. See below.

## **iPPS Command Switches**

Base switch. Specifies the number base to be used.

- Y - Binary
- O - Octal
- Q - Octal
- T - Decimal
- H - Hexadecimal

Data switch. Specifies the form in which data is represented.

- F - Represent data in its False (complemented) form

File switch. Specifies the file format.

- 80 - 8080 Absolute Object or 8080 Hexadecimal ASCII
- 86 - 8086 Absolute Object or 8086 Hexadecimal ASCII
- 286 - iPPS output file format

Patch switch. Specifies that patched object files can be read.

- P - Allow for reading a patched object file.

## **iPPS Command Defaults**

Base switch. Initially hexadecimal (H), but can be changed by the INITIALIZE command. The current default base can be overridden by appending a single letter base identifier to the numeric value. In the absence of an appended letter, a numeric entry is interpreted according to the default.

Data switch. Always true (T) and is only changed to false for the duration of a specific command by entering F as the data switch for that command.

File switch. iPPS output file format.

Patch switch. The default for the Patch switch is always Off. The switch is only changed to On for the duration of a specific file read command by entering P as the Patch switch for that command.

*file*. Current system default device.

## iPPS Command Defaults

Devices		Defaults		
Source	Destination	<i>startaddr</i>	<i>endaddr</i>	<i>destaddr</i>
BUFFER	<i>file</i>	Buffer start address(3)	Buffer end address(4)	0
<i>file</i>	BUFFER	File's lowest address(1)	File's highest address(2) limited by buffer size(5)	0
BUFFER	PROM	Buffer start address(3)	Buffer end address(4) limited by PROM size (5)	0
PROM	BUFFER	0	PROM size - 1	0
<i>file</i>	PROM	File's lowest address(1)	File's highest address(2)	0
PROM	<i>file</i>	0	PROM size - 1	0
<i>file</i>	none	File's lowest address(1)	File's highest address(2)	none
Buffer	none	Buffer start address(3)	Buffer end address(4)	none
PROM	none	0	PROM size - 1	none

### NOTES:

- (1) File's lowest address is the lowest address encountered while reading the file.
- (2) File's highest address is the highest address encountered while reading the file.
- (3) The most recent command that changed the lower boundary of the Buffer determines the Buffer start address. The TYPE command always resets the Buffer start address to 0.
- (4) Buffer end address is determined by the following expression:  
(Buffer start address) + (Buffer size - 1).
- (5) The most recent TYPE command fixes the Buffer size to the selected PROM size. iPPS initializes with a default Buffer size of 8K. If a *file* to Buffer operation is limited by the Buffer size, the default *endaddr* is: *startaddr* + (Buffer size - 1).

## iPPS Console Commands

**ALTER.** Edit and re-execute previous command.

ALTER

**BLANKCHECK.** Check for unprogrammed PROM.

BLANKCHECK [PROM (*startaddr* [, *endaddr*])]

**COPY.** File to PROM.

COPY *file* [(*startaddr* [, *endaddr*])]

TO PROM [(*destaddr*) ] [F]  $\left[ \begin{matrix} 80 \\ 86 \\ 286 \end{matrix} \right]$  [P]

**COPY. PROM to file.**

COPY PROM [(startaddr[,endaddr])]  
TO file [(destaddr)] [F]

**COPY. Buffer to PROM.**

COPY BUFFER [(startaddr[,endaddr])]  
TO PROM [(destaddr)] [F]

**COPY. PROM to buffer.**

COPY PROM [(startaddr[,endaddr])]  
TO BUFFER [(destaddr)] [F]

**COPY. Buffer to file.**

COPY BUFFER [(startaddr[,endaddr])]  
TO file [(destaddr)] [F]

**COPY. File to buffer.**

COPY file [(startaddr[,endaddr])]  
TO BUFFER [(destaddr)] [F]  $\left[ \begin{matrix} 80 \\ 86 \\ 286 \end{matrix} \right]$  [P]

**DISPLAY. Display data on console.**

$\left\{ \begin{array}{l} \text{DISPLAY } \left\{ \begin{array}{l} \text{PROM} \\ \text{BUFFER} \end{array} \right\} [(startaddr[,endaddr])] [F] \left[ \begin{array}{l} Y \\ O \\ Q \\ T \\ H \end{array} \right] \\ \text{DISPLAY file } [(startaddr[,endaddr])] [F] \left[ \begin{array}{l} 80 \\ 86 \\ 286 \end{array} \right] \left[ \begin{array}{l} Y \\ O \\ Q \\ T \\ H \end{array} \right] [P] \end{array} \right\}$

**ESC. Terminate current command.**

ESC

**EXIT. Exit iPPS and return control to ISIS-PDS.**

EXIT

**FORMAT. Interactively format buffer data.**

$$\left\{ \begin{array}{l} \underline{\text{FORMAT}} \left\{ \begin{array}{l} \underline{\text{PROM}} \\ \underline{\text{BUFFER}} \end{array} \right\} [(startaddr[,endaddr])] [F] \\ \underline{\text{FORMAT file}} [(startaddr[,endaddr])] [F] \left[ \begin{array}{l} 80 \\ 86 \\ 286 \end{array} \right] [P] \end{array} \right\}$$

**HELP. Selectively displays help information.**

HELP [command keyword]

**INITIALIZE. Set default number base and file format.**

$$\underline{\text{INITIALIZE}} \left[ \begin{array}{l} \text{Y} \\ \text{O} \\ \text{Q} \\ \text{T} \\ \text{H} \end{array} \right] \left[ \begin{array}{l} 80 \\ 86 \\ 286 \end{array} \right]$$

**LOADDATA. Fill section of buffer with constant.**

LOADDATA BUFFER [(startaddr[,endaddr])]  
WITH [byteval] [F]

**MAP. Display buffer and file structure and iPPS status.**

$$\underline{\text{MAP}} [file] \left[ \begin{array}{l} 80 \\ 86 \\ 286 \end{array} \right] [P]$$

**OVERLAY. Check buffer data against stuck bits in PROM.**

OVERLAY BUFFER [(startaddr[,endaddr])]  
TO PROM [(destaddr)] [F]

**PRINT. Print data on printer.**

$$\left\{ \begin{array}{l} \underline{\text{PRINT}} \left\{ \begin{array}{l} \underline{\text{PROM}} \\ \underline{\text{BUFFER}} \end{array} \right\} [(startaddr[,endaddr])] [F] \left[ \begin{array}{l} \text{Y} \\ \text{O} \\ \text{Q} \\ \text{T} \\ \text{H} \end{array} \right] \\ \underline{\text{PRINT file}} [(startaddr[,endaddr])] [F] \left[ \begin{array}{l} 80 \\ 86 \\ 286 \end{array} \right] \left[ \begin{array}{l} \text{Y} \\ \text{O} \\ \text{Q} \\ \text{T} \\ \text{H} \end{array} \right] [P] \end{array} \right\}$$



**REPEAT.** Repeat full execution of previous command.

REPEAT

**SUBSTITUTE.** Examine and modify buffer data.

SUBSTITUTE *addr* [F]



**TYPE.** Select PROM device type.

TYPE [*PROM device number*]

**VERIFY.** Verify PROM data with buffer data.

VERIFY [ BUFFER [(*startaddr*,*endaddr*)]  
TO PROM [(*destaddr*)] [F]

**WORKFILES.** Specify temporary workfiles.

WORKFILES :*Fn*:

**Hexadecimal to Decimal Conversion**

Most Significant Byte				Least Significant Byte			
Digit 4		Digit 3		Digit 2		Digit 1	
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
7654		3210		7654		3210	
Byte				Byte			

### Base Conversions

Dec	Bln	Hex	Oct	Dec	Bln	Hex	Oct
0	0000 0000	00	000	51	0011 0011	33	063
1	0000 0001	01	001	52	0011 0100	34	064
2	0000 0010	02	002	53	0011 0101	35	065
3	0000 0011	03	003	54	0011 0110	36	066
4	0000 0100	04	004	55	0011 0111	37	067
5	0000 0101	05	005	56	0011 1000	38	070
6	0000 0110	06	006	57	0011 1001	39	071
7	0000 0111	07	007	58	0011 1010	3A	072
8	0000 1000	08	010	59	0011 1011	3B	073
9	0000 1001	09	011	60	0011 1100	3C	074
10	0000 1010	0A	012	61	0011 1101	3D	075
11	0000 1011	0B	013	62	0011 1110	3E	076
12	0000 1100	0C	014	63	0011 1111	3F	077
13	0000 1101	0D	015	64	0100 0000	40	100
14	0000 1110	0E	016	65	0100 0001	41	101
15	0000 1111	0F	017	66	0100 0010	42	102
16	0001 0000	10	020	67	0100 0011	43	103
17	0001 0001	11	021	68	0100 0100	44	104
18	0001 0010	12	022	69	0100 0101	45	105
19	0001 0011	13	023	70	0100 0110	46	106
20	0001 0100	14	024	71	0100 0111	47	107
21	0001 0101	15	025	72	0100 1000	48	110
22	0001 0110	16	026	73	0100 1001	49	111
23	0001 0111	17	027	74	0100 1010	4A	112
24	0001 1000	18	030	75	0100 1011	4B	113
25	0001 1001	19	031	76	0100 1100	4C	114
26	0001 1010	1A	032	77	0100 1101	4D	115
27	0001 1011	1B	033	78	0100 1110	4E	116
28	0001 1100	1C	034	79	0100 1111	4F	117
29	0001 1101	1D	035	80	0101 0000	50	120
30	0001 1110	1E	036	81	0101 0001	51	121
31	0001 1111	1F	037	82	0101 0010	52	122
32	0010 0000	20	040	83	0101 0011	53	123
33	0010 0001	21	041	84	0101 0100	54	124
34	0010 0010	22	042	85	0101 0101	55	125
35	0010 0011	23	043	86	0101 0110	56	126
36	0010 0100	24	044	87	0101 0111	57	127
37	0010 0101	25	045	88	0101 1000	58	130
38	0010 0110	26	046	89	0101 1001	59	131
39	0010 0111	27	047	90	0101 1010	5A	132
40	0010 1000	28	050	91	0101 1011	5B	133
41	0010 1001	29	051	92	0101 1100	5C	134
42	0010 1010	2A	052	93	0101 1101	5D	135
43	0010 1011	2B	053	94	0101 1110	5E	136
44	0010 1100	2C	054	95	0101 1111	5F	137
45	0010 1101	2D	055	96	0110 1111	60	140
46	0010 1110	2E	056	97	0110 0001	61	141
47	0010 1111	2F	057	98	0110 0010	62	142
48	0011 0000	30	060	99	0110 0011	63	143
49	0011 0001	31	061	100	0110 0100	64	144
50	0011 0010	32	062	101	0110 0101	65	145

### Base Conversions (Continued)

Dec	Bin	Hex	Oct	Dec	Bin	Hex	Oct
102	0110 0110	66	146	153	1001 1001	99	231
103	0110 0111	67	147	154	1001 1010	9A	232
104	0110 1000	68	150	155	1001 1011	9B	233
105	0110 1001	69	151	156	1001 1100	9C	234
106	0110 1010	6A	152	157	1001 1101	9D	235
107	0110 1011	6B	153	158	1001 1110	9E	236
108	0110 1100	6C	154	159	1001 1111	9F	237
109	0110 1101	6D	155	160	1010 0000	A0	240
110	0110 1110	6E	156	161	1010 0001	A1	241
111	0110 1111	6F	157	162	1010 0010	A2	242
112	0111 0000	70	160	163	1010 0011	A3	243
113	0111 1110	71	161	164	1010 0100	A4	244
114	0111 0010	72	162	165	1010 0101	A5	245
115	0111 0011	73	163	166	1010 0110	A6	246
116	0111 0100	74	164	167	1010 0111	A7	247
117	0111 0101	75	165	168	1010 1000	A8	250
118	0111 0110	76	166	169	1010 1001	A9	251
119	0111 0111	77	167	170	1010 1010	AA	252
120	0111 1000	78	170	171	1010 1011	AB	253
121	0111 1001	79	171	172	1010 1100	AC	254
122	0111 1010	7A	172	173	1010 1101	AD	255
123	0111 1011	7B	173	174	1010 1110	AE	256
124	0111 1100	7C	174	175	1010 1111	AF	257
125	0111 1101	7D	175	176	1011 0000	B0	260
126	0111 1110	7E	176	177	1011 0001	B1	261
127	0111 1111	7F	177	178	1011 0010	B2	262
128	1000 0000	80	200	179	1011 0011	B3	263
129	1000 0001	81	201	180	1011 0100	B4	264
130	1000 0010	82	202	181	1011 0101	B5	265
131	1000 0011	83	203	182	1011 0110	B6	266
132	1000 0100	84	204	183	1011 0111	B7	267
133	1000 0101	85	205	184	1011 1000	B8	270
134	1000 0110	86	206	185	1011 1001	B9	271
135	1000 0111	87	207	186	1011 1010	BA	272
136	1000 1000	88	210	187	1011 1011	BB	273
137	1000 1001	89	211	188	1011 1100	BC	274
138	1000 1010	8A	212	189	1011 1101	BD	275
139	1000 1011	8B	213	190	1011 1110	BE	276
140	1000 1100	8C	214	191	1011 1111	BF	277
141	1000 1101	8D	215	192	1100 0000	C0	300
142	1000 1110	8E	216	193	1100 0001	C1	301
143	1000 1111	8F	217	194	1100 0010	C2	302
144	1001 0000	90	220	195	1100 0011	C3	303
145	1001 0001	91	221	196	1100 0100	C4	304
146	1001 0010	92	222	197	1100 0101	C5	305
147	1001 0011	93	223	198	1100 0110	C6	306
148	1001 0100	94	224	199	1100 0111	C7	307
149	1001 0101	95	225	200	1100 1000	C8	310
150	1001 0110	96	226	201	1100 1001	C9	311
151	1001 0111	97	227	202	1100 1010	CA	312
152	1001 1000	98	230	203	1100 1011	CB	313

### Base Conversions (Continued)

Dec	Bin	Hex	Oct	Dec	Bin	Hex	Oct
204	1100 1100	CC	314	230	1110 0110	E6	346
205	1100 1101	CD	315	231	1110 0111	E7	347
206	1100 1110	CE	316	232	1110 1000	E8	350
207	1100 1111	CF	317	233	1110 1001	E9	351
208	1101 0000	D0	320	234	1110 1010	EA	352
209	1101 0001	D1	321	235	1110 1011	EB	353
210	1101 0010	D2	322	236	1110 1100	EC	354
211	1101 0011	D3	323	237	1110 1101	ED	355
212	1101 0100	D4	324	238	1110 1110	EE	356
213	1101 0101	D5	325	239	1110 1111	EF	357
214	1101 0110	D6	326	240	1111 0000	F0	360
215	1101 0111	D7	327	241	1111 0001	F1	361
216	1101 1000	D8	330	242	1111 0010	F2	362
217	1101 1001	D9	331	243	1111 0011	F3	363
218	1101 1010	DA	332	244	1111 0100	F4	364
219	1101 1011	DB	333	245	1111 0101	F5	365
220	1101 1100	DC	334	246	1111 0110	F6	366
221	1101 1101	DD	335	247	1111 0111	F7	367
222	1101 1110	DE	336	248	1111 1000	F8	370
223	1101 1111	DF	337	249	1111 1001	F9	371
224	1110 0000	E0	340	250	1111 1010	FA	372
225	1110 0001	E1	341	251	1111 1011	FB	373
226	1110 0010	E2	342	252	1111 1100	FC	374
227	1110 0011	E3	343	253	1111 1101	FD	375
228	1110 0100	E4	344	254	1111 1110	FE	376
229	1110 0101	E5	345	255	1111 1111	FF	377

## ASCII Code List

Decimal	Hexadecimal	Character	Decimal	Hexadecimal	Character
0	00	NUL	56	38	8
1	01	SOH	57	39	9
2	02	STX	58	3A	:
3	03	ETX	59	3B	;
4	04	EOT	60	3C	<
5	05	ENQ	61	3D	=
6	06	ACK	62	3E	>
7	07	BEL	63	3F	?
8	08	BS	64	40	@
9	09	HT	65	41	A
10	0A	LF	66	42	B
11	0B	VT	67	43	C
12	0C	FF	68	44	D
13	0D	CR	69	45	E
14	0E	SO	70	46	F
15	0F	SI	71	47	G
16	10	DLE	72	48	H
17	11	DC1	73	49	I
18	12	DC2	74	4A	J
19	13	DC3	75	4B	K
20	14	DC4	76	4C	L
21	15	NAK	77	4D	M
22	16	SYN	78	4E	N
23	17	ETB	79	4F	O
24	18	CAN	80	50	P
25	19	EM	81	51	Q
26	1A	SUB	82	52	R
27	1B	ESC	83	53	S
28	1C	FS	84	54	T
29	1D	GS	85	55	U
30	1E	RS	86	56	V
31	1F	US	87	57	W
32	20	SP	88	58	X
33	21	!	89	59	Y
34	22	"	90	5A	Z
35	23	#	91	5B	[
36	24	\$	92	5C	\
37	25	%	93	5D	]
38	26	&	94	5E	^
39	27	'	95	5F	_
40	28	(	96	60	`
41	29	)	97	61	a
42	2A	*	98	62	b
43	2B	+	99	63	c
44	2C	,	100	64	d
45	2D	-	101	65	e
46	2E	.	102	66	f
47	2F	/	103	67	g
48	30	0	104	68	h
49	31	1	105	69	i
50	32	2	106	6A	j
51	33	3	107	6B	k
52	34	4	108	6C	l
53	35	5	109	6D	m
54	36	6	110	6E	n
55	37	7	111	6F	o

### ASCII Code List (Continued)

Decimal	Hexadecimal	Character	Decimal	Hexadecimal	Character
112	70	p	120	78	x
113	71	q	121	79	y
114	72	r	122	7A	z
115	73	s	123	7B	
116	74	t	124	7C	
117	75	u	125	7D	
118	76	v	126	7E	~
119	77	w	127	7F	DEL

### ASCII Code Definition

Abbreviation	Meaning	Decimal Code
NUL	NULL Character	0
SOH	Start of Heading	1
STX	Start of Text	2
ETX	End of Text	3
EOT	End of Transmission	4
ENQ	Enquiry	5
ACK	Acknowledge	6
BEL	Bell	7
BS	Backspace	8
HT	Horizontal Tabulation	9
LF	Line Feed	10
VT	Vertical Tabulation	11
FF	Form Feed	12
CR	Carriage Return	13
SO	Shift Out	14
SI	Shift In	15
DLE	Data Link Escape	16
DC1	Device Control 1	17
DC2	Device Control 2	18
DC3	Device Control 3	19
DC4	Device Control 4	20
NAK	Negative Acknowledge	21
SYN	Synchronous Idle	22
ETB	End of Transmission Block	23
CAN	Cancel	24
EM	End of Medium	25
SUB	Substitute	26
ESC	Escape	27
FS	File Separator	28
GS	Group Separator	29
RS	Record Separator	30
US	Unit Separator	31
SP	Space	32
DEL	Delete	127

## ASCII Code in Binary

	MSB		0	1	2	3	4	5	6	7
	LSB	000	001	010	011	100	101	110	111	
0	0000	NUL	DLE	SP	0	@	P	`	p	
1	0001	SOH	DC1	!	1	A	Q	a	q	
2	0010	STX	DC2	"	2	B	R	b	r	
3	0011	ETX	DC3	#	3	C	S	c	s	
4	0100	EOT	DC4	\$	4	D	T	d	t	
5	0101	ENQ	NAK	%	5	E	U	e	u	
6	0110	ACK	SYN	&	6	F	V	f	v	
7	0111	BEL	ETB	'	7	G	W	g	w	
8	1000	BS	CAN	(	8	H	X	h	x	
9	1001	HT	EM	)	9	I	Y	i	y	
A	1010	LF	SUB	*	:	J	Z	j	z	
B	1011	VT	ESC	+	;	K	[	k		
C	1100	FF	FS	,	<	L	\	l		
D	1101	CR	GS	-	=	M	]	m	~	
E	1110	SO	RS	.	>	N	^	n	o	
F	1111	SI	VS	/	?	O	_	o	DEL	

## Control Codes

Character	Code in Hexadecimal	Character	Code in Hexadecimal
CTRL-@	00	CTRL-P	10
CTRL-A	01	CTRL-Q	11
CTRL-B	02	CTRL-R	12
CTRL-C	03	CTRL-S	13
CTRL-D	04	CTRL-T	14
CTRL-E	05	CTRL-U	15
CTRL-F	06	CTRL-V	16
CTRL-G	07	CTRL-W	17
CTRL-H	08	CTRL-X	18
CTRL-I	09	CTRL-Y	19
CTRL-J	0A	CTRL-Z	1A
CTRL-K	0B	CTRL-[	1B
CTRL-L	0C	CTRL-\	1C
CTRL-M	0D	CTRL-]	1D
CTRL-N	0E	CTRL-^	*
CTRL-O	0F	CTRL-~	**

\* - Use ↑ to generate the code 1EH.

\*\* - Use ← to generate the code 1FH.

The IPDS CRT responds to CTRL-G (ASCII Bell), CTRL-H (ASCII Backspace), CTRL-J (ASCII Linefeed), CTRL-M (ASCII Carriage Return), CTRL-[ (ASCII Escape).

## Function Codes

FUNC-A thru FUNC-Z\*. ASCII code for A-Z with MSB set.  
FUNC-0 thru FUNC-9. ASCII code for 0-9 with MSB set.

\*FUNC-S, FUNC-R, and FUNC-T are not sent to the processor.

FUNC-a thru FUNC-z. Same as FUNC-A thru FUNC-Z.

The codes for other FUNC characters are undefined except for FUNC-HOME, FUNC-↑, and FUNC-↓ which are used in dual processor systems.

## Escape Sequences

ESC, G. Enter graphics mode. Any codes from 00H-1FH (except 02H and 1BH) produce a graphics symbol as shown in the following chart. Codes from 20H to 7FH produce the corresponding ASCII character.

ESC, N. Return from graphics mode.

ESC, A. Move the cursor up one line.

ESC, B. Move the cursor down one line.

ESC, C. Move the cursor to the right one character.

ESC, D. Move the cursor to the left one character.

ESC, E. Home the cursor and clear the screen.

ESC, H. Home the cursor.

ESC, J. Erase from the current cursor location to the end of the screen.

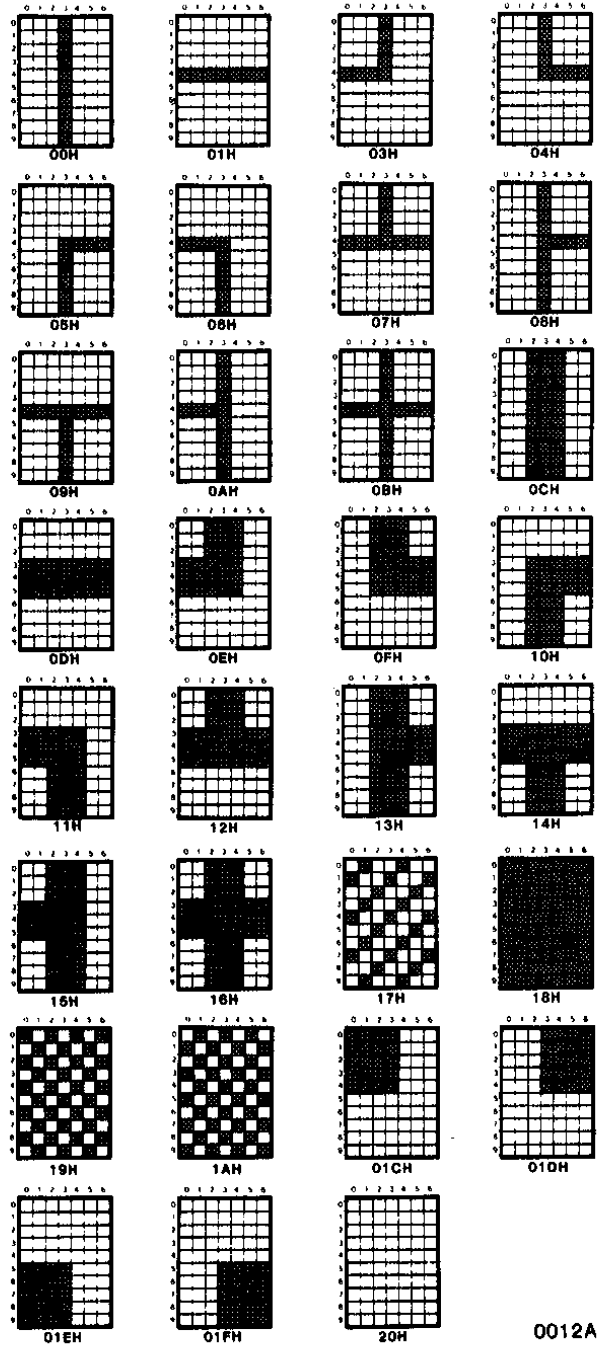
ESC, K. Erase the line containing the cursor.

ESC, Y  $x, y$ . Move the cursor to the address specified by the third ( $x$ ) and fourth ( $y$ ) bytes.

## Graphics Codes

The following chart shows the graphics symbols available and the corresponding hexadecimal value that creates that symbol when output to the CRT screen when in graphics mode. The codes 02H and 1BH do not produce graphics symbols because they are reserved for the alternate escape and escape characters used in the escape sequences described previously.





**Powers of Two**

$2^n$	n
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16
131 072	17
262 144	18
524 288	19
1 048 576	20
2 097 152	21
4 194 304	22
8 388 608	23
16 777 216	24

**Conversion Between Powers of 2 and 16**

$2^m = 16^n$
$2^0 = 16^0$
$2^4 = 16^1$
$2^8 = 16^2$
$2^{12} = 16^3$
$2^{16} = 16^4$
$2^{20} = 16^5$
$2^{24} = 16^6$
$2^{28} = 16^7$
$2^{32} = 16^8$
$2^{36} = 16^9$
$2^{40} = 16^{10}$
$2^{44} = 16^{11}$
$2^{48} = 16^{12}$
$2^{52} = 16^{13}$
$2^{56} = 16^{14}$
$2^{60} = 16^{15}$
$2^{64} = 16^{16}$

**Powers of Sixteen**

$16^n$	n
1	0
16	1
256	2
4 096	3
65 536	4
1 048 576	5
16 777 216	6
268 435 456	7
4 294 967 296	8
68 719 476 736	9
1 099 511 627 776	10
17 592 186 044 416	11
281 474 976 710 656	12
4 503 599 627 370 496	13
72 057 594 037 927 936	14
1 152 921 504 606 846 976	15



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