

CP/M-86®  
OPERATING SYSTEM  
COMMAND SUMMARY

Copyright © 1982

Digital Research  
P.O. Box 579  
160 Central Avenue  
Pacific Grove, CA 93950  
(408) 649-3896  
TWX 910 360 5001

All Rights Reserved

# TABLE OF CONTENTS

## COPYRIGHT

Copyright © 1982 by Digital Research. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of Digital Research, Post Office Box 579, Pacific Grove, California, 93950.

## DISCLAIMER

Digital Research makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, Digital Research reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Digital Research to notify any person of such revision or changes.

## TRADEMARKS

CP/M and CP/M-86 are registered trademarks of Digital Research. ASM-86, CBASIC-86, CP/M-80, and DDT-86 are trademarks of Digital Research. Intel is a registered trademark of Intel Corporation.

The CP/M-86 Operating System Command Summary was prepared using the Digital Research TEX Text Formatter and printed in the United States of America by Commercial Press/Monterey.

\*\*\*\*\*  
\* First Revision: May 1982 \*  
\* First Printing: May 1982 \*  
\*\*\*\*\*

How to enter a CP/M-86 Command . . . . .	1
CP/M-86 File Specifications . . . . .	1
Command Summary Conventions . . . . .	2
ASM86 . . . . .	3
COPYDISK . . . . .	4
DDT86 . . . . .	5
DIR . . . . .	7
DIRS . . . . .	8
ED . . . . .	9
ERA . . . . .	12
GENCMD . . . . .	13
HELP . . . . .	14
PIP . . . . .	15
REN . . . . .	17
STAT . . . . .	18
SUBMIT . . . . .	20
TOD . . . . .	21
TYPE . . . . .	22
USER . . . . .	23
CP/M-86 Control Character Summary . . . . .	24
CP/M-86 Filetypes . . . . .	25
CP/M-86 Messages . . . . .	26
CP/M-86 BDOS Function Summary . . . . .	45
CP/M-86 File Control Block . . . . .	47
CP/M-86 Base Page Values . . . . .	48

## HOW TO ENTER A CP/M-86 COMMAND

---

To give CP/M-86<sup>®</sup> a command, you must type a complete command line following the CP/M-86 system prompt. A CP/M-86 command line is composed of a command, an optional command tail, and a carriage return. The command identifies a command (program) to be executed. The optional command tail can consist of a drive specification, one or more file specifications, and some options or parameters. To complete the command, you must press the carriage return key.

## CP/M-86 FILE SPECIFICATIONS

---

CP/M-86 identifies every file by its unique file specification. A file specification can have three parts: a drive specification, a primary filename, and a filetype, all separated by their appropriate delimiters. A drive letter must be followed by a colon. A filetype must be preceded by a period. The term filespec is an abbreviation for file specification, and indicates any valid combination of the drive specification, the filename and the filetype. This summary uses the following symbols to designate the parts of a filespec.

**d:** represents the optional drive specification, which can be a single alphabetical character in the range A through P followed by a colon.

**filename** represents the required primary filename, which can be 1 to 8 alphabetic or numeric characters.

**typ** represents the optional filetype, which can be 0 to 3 alphabetic or numeric characters separated from the primary filename by a period.

Valid combinations of the elements of a filespec are shown below:

- filename
- d:filename
- filename.typ
- d:filename.typ

Certain CP/M-86 commands can select and process several files when wildcard characters are included in the primary filename or filetype. The two wildcard characters are ?, which matches any single letter in the same position, and \*, which matches any character at that position and any other characters remaining in the filename or filetype. The command summary indicates which commands can accept wildcard characters (\* or ?) in a filename or filetype.

## COMMAND SUMMARY CONVENTIONS

The command summary lists each CP/M-86 command in alphabetical order using the following special symbols to define command syntax:

Symbol	Meaning
{ }	indicate an optional item.
	separates choices.
n	indicates a number.
<cr>	indicates a carriage return keystroke.
↑	indicates a control keystroke.
o	indicates an option or option list.
RW	means Read Write.
RO	means Read Only.
SYS	means System attribute.
DIR	means Directory attribute.
...	preceding element can be repeated as many times as desired.
*	wildcard- replaces all or part of a filename and/or filetype.
?	wildcard- replaces any single character in the same position of a filename or filetype.

## ASM86

### Syntax:

ASM86 filespec { \$Ad Hd Pd Sd Fd }

A - source file drive .A86  
d - logical drives A-P

H - hex file drive .H86  
d - logical drives A-P, X,Y,Z

P - print file drive .LST  
d - logical drives A-P, X,Y,Z

S - symbol file drive .SYM  
d - logical drives A-P, X,Y,Z

F - format of hex file .H86  
d - D=Digital Research, I=Intel®

### Purpose:

Assembles assembly language statements, producing an object file in hexadecimal format, a print file, and a symbol table file. The assumed filetype of source and included files is A86. The special characters X, Y, and Z indicate output to the console, output to the printer, and zero output respectively. If no format is specified, Digital Research format (FD) is assumed.

### Examples:

```
A>ASM86 PROG
A>ASM86 PROG $$Z PX
A>ASM86 PROG $HB PY AC
```

## COPYDISK

**Syntax:**

COPYDISK

**Purpose:**

Copies all information on one disk to another disk, including the CP/M-86 system tracks if they are present on the source disk. COPYDISK prompts you for the source and destination drives.

**Example:**

A&gt;COPYDISK

## DDT86

**Syntax:**

DDT86 {filespec}

**Purpose:**

Aids debugging of 8086 and 8088 programs. Assumes filetype of CMD if only a filename is specified. Use GENCMD to convert hexadecimal (H86) files to command (CMD) files before using DDT-86<sup>TM</sup>.

**Examples:**

```
A>DDT86
A>DDT86 PROGRAM1
A>DDT86 PROGRAM2.CMD
A>DDT86 B:PROGRAM3.CMD
```

**DDT-86 Command Summary**

The command character can be followed by one or more arguments, which can be hexadecimal values, file specifications, or other information, depending on the command. Arguments are separated from each other by commas or spaces. No spaces are allowed between the command character and the first argument.

Command	Action
<b>As</b>	(Assemble) Enter Assembly Language Statements
<b>Bs,f,sl</b>	(Block Compare) Compare Blocks of Memory
<b>D{w}{s{,f}}</b>	(Display) Display Memory in Hex and ASCII
<b>Efilespec</b>	(Execution) Load Program for Execution
<b>Fs,f,bc</b>	(Fill) Fill Memory Block - Byte
<b>Fws,f,wc</b>	(Fill) Fill Memory Block - Word
<b>G{s}{,bl{,b2}}</b>	(Go) Begin Execution

## DDT-86 Command Summary (continued)

Command	Action
<b>Hwcl,wc2</b>	(Hex) Hexadecimal Sum and Difference
<b>Icommand tail</b>	(Input) Set Up Input Command Line
<b>L{s,f}</b>	(List) List Memory in Mnemonic Form
<b>Ms,f,d</b>	(Move) Move Memory Block
<b>Rfilespec</b>	(Read) Read Disk File to Memory
<b>S{W}s</b>	(Set) Set Memory Values
<b>T{n}</b>	(Trace) Trace Program Execution
<b>TS{n}</b>	(Trace) Trace and Show All Registers
<b>U{n}</b>	(Untrace) Monitor Execution without Trace
<b>US{n}</b>	(Untrace) Monitor and Show All Registers
<b>V</b>	(Verify) Show Memory Layout after Disk Read
<b>Wfilespec{s,f}</b>	(Write) Write Content of Block to Disk
<b>X{r}</b>	(Examine) Examine and Modify CPU Registers

Symbol	Meaning
bc	byte constant
b1	breakpoint one
b2	breakpoint two
d	destination for data
f	final address
n	number of instructions to execute
r	register or flag name
s	starting address
s1	second starting address
W	word 16-bit
wc	word constant

## DIR

## Syntax:

DIR {filespec}

## Purpose:

Displays the names of nonsystem (DIR) files in the directory of an on-line disk. Accepts wildcards in the filename and filetype.

## Examples:

```
A>DIR
A>DIR B:
A>DIR C:MYFILE.DAT
A>DIR *.CMD
A>DIR A*.A86
A>DIR PROG???.H86
A>DIR PROGRAM.*
```

## DIRS

## Syntax:

DIRS {filespec}

## Purpose:

Displays the names of system (SYS) files in the directory of an on-line disk. Accepts wildcards in the filename and filetype.

## Examples:

```
A>DIRS
A>DIRS B:
A>DIRS C:MYFILE.COM
A>DIRS *.COM
A>DIRS A*.COM
A>DIRS PROG???.COM
A>DIRS PROGRAM.*
```

## ED

## Syntax:

ED input-filespec {d:|output-filespec}

## Purpose:

Edits source program or text files. Redirect or rename the new version of the file by specifying the destination drive or destination filespec.

## Examples:

```
A>ED TEST.DAT
A>ED TEST.DAT B:
A>ED TEST.DAT TEST2.DAT
A>ED TEST.DAT B:TEST2.DAT
```

## ED Command Summary

Command	Action
nA	append n lines from original file to memory buffer
OA	append file until buffer is one half full
#A	append file until buffer is full or come to end of file
B, -B	move CP to the beginning (B) or bottom (-B) of buffer
nC, -nC	move CP n characters forward (C) or back (-C) through buffer
nD, -nD	delete n characters before (-D) or from (D) the CP
E	save new file and return to CP/M-86
Fstring{↑z}	find character string

## ED Command Summary (continued)

Command	Action
<b>H</b>	save the new file, then reedit, using the new file as the original file
<b>I</b>	enter insert mode; use ↑Z to exit insert mode
<b>Istring{↑z}</b>	insert string at CP
<b>Jsearch_str↑zins_str↑zdel_to_str{↑z}</b>	juxtapose strings
<b>nK, -nK</b>	delete (kill) n lines from the CP
<b>nL, -nL, OL</b>	move CP n lines
<b>nMcommands</b>	execute commands n times
<b>n, -n</b>	move CP n lines and display that line
<b>n:</b>	move to line n
<b>:ncommand</b>	execute command through line n
<b>Nstring{↑z}</b>	extended find string
<b>O</b>	return to original file
<b>nP, -nP</b>	move CP 23 lines forward and display 23 lines at console
<b>Q</b>	abandon new file, return to CP/M-86
<b>R</b>	read X\$\$\$\$\$\$\$.LIB file into buffer
<b>Rfilespec{↑z}</b>	read filespec into buffer
<b>Sdelete string↑zinsert string{↑z}</b>	substitute string

## ED Command Summary (continued)

Command	Action
<b>nT, -nT, OT</b>	type n lines
<b>U, -U</b>	upper-case translation
<b>V, -V, OV</b>	line numbering on/off, display free buffer space
<b>nW</b>	write n lines to new file
<b>nX</b>	write n lines to X\$\$\$\$\$\$\$.LIB
<b>nXfilespec{↑z}</b>	write n lines to filespec
<b>OX</b>	delete file X\$\$\$\$\$\$\$.LIB
<b>OXfilespec{↑z}</b>	delete filespec
<b>nZ</b>	wait n seconds

**Note:** CP refers to the character pointer.  
CP points to the current character being  
referenced in the edit buffer.



## ERA

**Syntax:**

ERA filespec

**Purpose:**

Erases a file or group of files. Accepts wildcards in the filename and filetype.

**Examples:**

```
A>ERA DRAFT.BAK
A>ERA B:LETTER.DAT
A>ERA C:LETTER.*
A>ERA D:*.BAK
A>ERA B:*.*
```

## GENCMD

**Syntax:**

GENCMD filespec { 8080  
CODE [An,Bn,Mn,Xn]  
DATA [An,Bn,Mn,Xn]  
STACK [An,Bn,Mn,Xn]  
EXTRA [An,Bn,Mn,Xn] }

A - Absolute memory location  
B - Beginning address of group  
in .H86 file  
M - Minimum memory required  
X - Maximum memory required

**Purpose:**

Converts hexadecimal object file (filetype assumed to be .H86) into executable file (of type .CMD). Switches controlling the type of memory model and group addresses can be included. All addresses are paragraph addresses. All values must be in hexadecimal.

**Examples:**

```
A>GENCMD PROG
B>GENCMD PROG1 8080
B>GENCMD PROG2 DATA[M20]
A>GENCMD PROG3 DATA[B4C,M260,XFFF]
A>GENCMD PROG4 CODE[A40] DATA[M30]
```

## HELP

**Syntax:**

HELP {topic}{,subtopic}...{[P]}

**Purpose:**

Supplies information on how to use CPM-86 commands. HELP with no command tail lists the available topics. HELP with a list of topics and subtopics displays information about that command on your screen. HELP allows up to 8 levels of subtopics. When the P option is used, the HELP display does not pause every 23 lines.

**Examples:**

```
A>HELP
A>HELP dir
A>HELP stat options[P]
HELP>help
```

## PIP

**Syntax:**

Destination	Source
-------------	--------

```
PIP d:[Gn]}filespec{[Gn]=filespec{o}}
PIP filespec{[Gn]}=d:[Gn]}
PIP {d:}filename{.typ}=filespec1,filespec2,{,..}
PIP filespec{[Gn]}|dev:=filespec{o}}|dev:{o}}
```

**Purpose:**

Copies files, combines files, and transfers files between disks and peripheral devices. The first filespec is the destination. The second filespec is the source. The source or destination can be any CP/M-86 logical device. PIP with no command tail displays an \* prompt and awaits your series of commands, entered and executed one line at a time. Source filespecs with options can be repeated, separated by commas, to combine two or more files into one file. The [o] is any combination of the available options. The [Gn] option in the destination designates USER number n.

**Examples:**Copy from one disk to another:

```
A>PIP B:=A:DRAFT.TXT
A>PIP B:DRAFT.TXT=A:
A>PIP B:[G5]=NEWDRAFT.TXT
A>PIP NEWDRAFT.TXT=B:[G5]
```

Copy a file and rename it:

```
A>PIP B:NEWDRAFT.TXT=A:OLDDRAFT.TXT
A>PIP NEWDRAFT.TXT=OLDDRAFT.TXT
```

Copy multiple files:

```
A>PIP <cr>
A>PIP B:=*.TXT[V]
A>PIP B:=*.CMD[R]
A>PIP B:=C:DRAFT.*[W]
A>PIP B:=*.*
A>PIP B:=C:*.*
```

Combine multiple files:

```
A>PIP B:NEW.DAT=FILE1.DAT,FILE2.DAT
```

## PIP (continued)

Copy, rename, and place in user 1:

A>PIP NEWDRAFT.TXT[G1]=OLDDRAFT.TXT

Copy, rename, and get from user 1:

A>PIP NEWDRAFT.TXT=OLDDRAFT.TXT[G1]

Copy to/from logical devices:

A>PIP B:FUNFILE.SUB=CON:

A>PIP LST:=CON:

A>PIP LST:=B:DRAFT.TXT[T8]

A>PIP PRN:=B:DRAFT.TXT

## Options:

Option	Description
Dn	Delete any characters past column n.
E	Echo transfer to console.
F	Filter form-feeds from source data.
Gn	Get from or go to user code n. (default n=current user num.)
H	Test for valid Hex format.
I	Ignore :00 Hex data records and test for valid Hex format.
K	Kill display of filenames on console.
L	Translate upper-case to lower-case.
N	Number output lines
O	Object file transfer; ↑Z ignored.
Pn	Insert form-feeds every n lines.
Qs↑Z	Quit copying from source at string s.
R	Read files that have been set to SYStem.
Ss↑Z	Start copying from the source at the string s.
Tn	Expand tabs to n spaces.
U	Translate lower-case to upper-case.
V	Verify that data has been written correctly.
W	Write over Read Only files without console query.
Z	Zero the parity bit.

All options except G,K,O,R,V, and W force an ASCII file transfer (character by character, terminated by a ↑Z).

## REN

## Syntax:

REN {d:}newname{.typ}=oldname{.typ}

## Purpose:

Changes the name of the existing file specified by oldname to the name specified by newname. You cannot specify two different drives.

## Examples:

A>REN NEWFILE.DAT=OLDFILE.DAT

A>REN B:NEWFILE.DAT=OLDFILE.DAT

A>REN B:NEWLIST=OLDLIST

## STAT

## STAT (continued)

## Syntax:

```

STAT
STAT d:=RO
STAT filespec {RO|RW|SYS|DIR|SIZE}
STAT {d:}DSK:|{d:}USR:|DEV:|VAL:
STAT logical device:= physical device:

```

## Purpose:

Supplies information about the disk drives, files, and peripheral devices attached to the computer. STAT also changes attributes of files and devices. STAT with no command tail returns the amount of free storage space in kilobytes (1024 bytes or 1K) for all logged-in disks since CP/M-86 was started or reloaded. It also indicates whether the drive is set to Read Only (RO) or Read Write (RW). Drives are in a RW state by default, and become RO when you change a disk and do not enter a ↑C. Use STAT to set the drive to RO and ↑C to set it to RW. STAT with a file specified in the command tail returns the number of kilobytes used by that file. STAT with a filespec can also set a file or group of files to RO, RW, SYS, or DIR. STAT DSK: shows the characteristics of the default or specified drive. STAT USR: shows which user numbers contain files on a given disk. STAT VAL: shows the possible external devices that can be assigned to your computer. STAT accepts wildcards in the command tail.

Note that STAT accepts a dollar sign \$, square bracket [, or no delimiter before the RO, RW, SYS, DIR, and SIZE settings.

## Options:

```

RW      Read Write
RO      Read Only
SYS     System attribute
DIR     NO System attribute
SIZE    Show the size of the
         file or files
VAL:    Shows possible devices
USR:    Shows USER numbers
         containing files
DSK:    Shows characteristics
         of the default drive
DEV:    Shows current physical to
         logical device assignments

```

## Examples:

```

A>STAT
A>STAT B:=RO
A>STAT myfile.txt
A>STAT C:letter.*
A>STAT genledgr.dat RO,SYS
A>STAT *.cmd RO
A>STAT *.bak
A>STAT B:*. *

```

**SUBMIT****Syntax:**

SUBMIT filespec {actual parameters}

**Purpose:**

Submits a batch process consisting of a file of CP/M-86 commands (one command per line in file). The SUBMIT file must have a filetype of SUB. Any optional parameters (for example a drive or file specification) following the file specification in the command line are substituted for their corresponding formal parameters (\$1,\$2,\$3...) in the file of type SUB.

**Examples:**

A>SUBMIT START  
 A>SUBMIT B:START  
 A>SUBMIT START C: LETTER

**TOD****Syntax:**

TOD {mm/dd/yy hh:mm:ss | P}

**Purpose:**

Display or set the date and time. TOD with no command tail displays the system date and time. Time is represented as a twenty-four hour clock, with hour values from 00 to 11 for the morning, and 12 to 23 for the afternoon and evening. TOD with the P option displays the date and time continuously, until you press any key. TOD sets the date and time to that which is specified in the command tail. When you start CP/M-86, these values are initially set to zero. Use TOD to set the current date and time. Supply a two-digit representation of the month, day, and year for mm, dd, and yy respectively. Supply a two-digit representation of the hours, minutes, and seconds for hh, mm, and ss respectively. When the time specified corresponds to the correct time, strike a key, and TOD sets the current date and time.

**Examples:**

A>TOD  
 A>TOD P  
 A>TOD 02/14/81 12:00:00

## TYPE

**Syntax:**

TYPE filespec

**Purpose:**

Displays contents of an ASCII file on the screen. Press any key to discontinue the display. TYPE does not accept wildcard filespecs. Entering a ↑P prior to the type command causes the output to be echoed to the printer until you enter another ↑P.

**Examples:**

A>TYPE letter.dat  
 B>TYPE a:document.law  
 C>TYPE program.bas  
 D>TYPE program.a86

## USER

**Syntax:**

USER n  
 n = 0-15

**Purpose:**

Displays and changes the current user number. USER with no command tail displays the current user number. USER with a number from 0 to 15 changes the current user number to the number specified by n. CP/M-86 assumes a default user number of 0. Files set to SYSTEM on USER 0 are available to all USER areas as Read Only.

**Examples:**

A>USER  
 B>USER 2  
 A>USER 7

## CP/M-86 CONTROL CHARACTER SUMMARY

Keystroke	Action
<b>BACKSPACE</b>	moves cursor back one space; erases previous character.
<b>↑C</b>	prompts to abort a program currently running at a given console.
<b>DEL</b>	same as RUB.
<b>↑E</b>	forces a physical carriage return, but does not send command to CP/M-86.
<b>↑H</b>	same as BACKSPACE.
<b>↑J</b>	line-feed, terminates input at the console.
<b>↑M</b>	same as carriage return.
<b>↑P</b>	echoes all console activity at the printer; a second ↑P ends printer echo. This only works if your system is connected to a printer.
<b>↑R</b>	retypes current command line; useful after using RUB or DEL key.
<b>RETURN</b>	carriage return.
<b>RUB</b>	deletes character to the left of cursor; echoes character deleted - cursor moves right.
<b>↑S</b>	stops console listing temporarily; ↑S resumes the listing.
<b>↑U</b>	cancels line, displays #, cursor moves down one line and awaits a new command.
<b>↑X</b>	deletes all characters in command line.
<b>↑Z</b>	string or field separator.

## CP/M-86 FILETYPES

CP/M-86 identifies every file by a unique file specification, which consists of a drive specifier, a filename, and a filetype. The filetype is an optional three character ending separated from the filename by a period. The filetype generally indicates a special kind of file. The common filetypes and their meanings are listed below.

Filetype	Indication
<b>A86</b>	Assembly language source file; ASM-86 <sup>TM</sup> , the CP/M-86 assembler, assembles or translates a file of type .A86 into machine language.
<b>BAK</b>	Back-up file created by a text editor; an editor renames the source file with this filetype to indicate that the original file has been processed. The original file stays on the disk as the back-up file, so you can refer to it.
<b>BAS</b>	CBASIC-86 <sup>TM</sup> program source file.
<b>CMD</b>	Command file that contains instructions in machine executable code.
<b>COM</b>	8080 executable file.
<b>H86</b>	Program file in hexadecimal format.
<b>INT</b>	CBASIC-86 program intermediate language file.
<b>LST</b>	Printable file that can be displayed on a console or printer.
<b>PRN</b>	Printable file that can be displayed on a console or printer.
<b>SUB</b>	Filetype required for SUBMIT input file containing one or more CP/M-86 commands. The SUBMIT program executes the commands in the file of type SUB providing a batch mode for CP/M-86.
<b>SYM</b>	Symbol table file.
<b>\$\$\$</b>	Temporary file created by PIP.

## CP/M-86 MESSAGES

## CP/M-86 Messages (continued)

Message	Meaning
---------	---------

**Ambiguous operand**

DDT-86. An attempt was made to assemble a command with an ambiguous operand. Precede the operand with the prefix BYTE or WORD.

**Bad Directory on d:  
Space Allocation Conflict:  
User n d:filename.typ**

STAT. This message is followed by a list of one or more filespecs. The files listed contain data blocks that are already allocated to another file on the disk. The error can be caused by a hardware or software failure. Correct the error by erasing the files displayed and rebooting CP/M-86. Note that if CP/M-86 is not rebooted after erasing the files, the problem reappears.

**BDOS err on d:**

CP/M-86 replaces d: with the drive specification of the drive where the error occurred. This message appears when CP/M-86 finds no disk in the the drive, when the disk is improperly formatted, when the drive latch is open, or when power to the drive is off. Check for one of these situations and retry.

**BDOS err on d: bad sector**

This could indicate a hardware problem or a worn or improperly formatted disk. Press ↑C to terminate the program and return to CP/M-86, or press the return key to ignore the error.

**BDOS err on d: file R/O**

An erase, rename, or set file attributes operation was attempted on a Read Only file. The file should first be set to Read Write (RW) with the command: STAT filespec RW.

**BDOS err on d: select**

CP/M-86 has received a request specifying a nonexistent drive, or disk in drive is improperly formatted. CP/M-86 terminates the current program as soon as you press any key.

**BDOS err on d: R/O**

Drive has been assigned Read Only status with a STAT command, or the disk in the drive has been changed without being initialized with a ↑C. CP/M-86 terminates the current program as soon as you press any key.

**Break "x" at c**

ED. x is one of the symbols described below and c is the command letter being executed when the error occurred.

# Search failure. ED cannot find the string specified in an F, S, or N command.

? Unrecognized command letter c. ED does not recognize the indicated command letter, or an E, H, Q, or O command is not alone on its command line.

O The file specified in an R command cannot be found.

> Buffer full. ED cannot put any more characters in the memory buffer, or the string specified in an F, N, or S command is too long.



## CP/M-86 Messages (continued)

E Command aborted. A keystroke at the console aborted command execution.

F Disk or directory full. This error is followed by either the disk or directory full message. Refer to the recovery procedures listed under these messages

**Cannot close****Cannot close file**

ASM-86. An output file cannot be closed. This is a fatal error that terminates ASM-86 execution. You should take appropriate action after checking to see if the correct disk is in the drive and that the disk is not write-protected.

DDT-86. The disk file written by a W command cannot be closed. This is a fatal error that terminates DDT-86 execution. You should take appropriate action after checking to see if the correct disk is in the drive and that the disk is not write-protected.

GENCMD. The CMD file written by a W command cannot be closed. This is a fatal error that terminates GENCMD execution. You should take appropriate action after checking to see if the correct disk is in the drive and that the disk is not write-protected.

SUBMIT. This error can occur during SUBMIT file processing. Check to see if the correct system disk is in the A drive and that the disk is not write-protected. The SUBMIT job can be restarted after rebooting CP/M-86.

## CP/M-86 Messages (continued)

**Cannot open source**

GENCMD. The hex file specified in the GENCMD command line cannot be found. The hex file must have the filetype H86. Check to see that the correct disk was specified and try again.

**Command Buffer Overflow**

SUBMIT. The SUBMIT buffer allows up to 2048 characters in the input file.

**Command too long**

SUBMIT. A command in the SUBMIT file cannot exceed 125 characters.

**Checksum error**

GENCMD. A hex record checksum error was encountered. The hex record that produced the error must be corrected, probably by recreating the hex file.

**Command name?**

If CP/M-86 cannot find the command you specified, it returns the command name you entered followed by a question mark. Check that you have typed the command name correctly, or that the command you requested exists as a .CMD file on the default or specified disk.

GENCMD. An invalid GENCMD command line was entered. If the command was mistyped then try again.

**DESTINATION IS R/O, DELETE (Y/N)?**

PIP. The destination file specified in a PIP command already exists and it is Read Only. If you type Y, the destination file is deleted before the file copy is done.

## CP/M-86 Messages (continued)

**Directory full**

ASM-86. There is not enough directory space for the output files. You should either erase some unnecessary files or get another disk with more directory space and execute ASM-86 again.

ED. There is not enough directory space for the file being written. You can use the OXfilespec command to erase any unnecessary files on the disk without leaving the editor. Alternatively, you can save the contents of the memory buffer on another disk with the command B#Xfilespec, where filespec is a file on a different drive. You can then quit the edit. If you reedit the file, the output should be placed on a different drive with the command ED filespec d:, where d: is a valid drive name other than the drive containing the source file. You can read the file saved with the Rfilespec command. Caution, part of the file might not be in the memory buffer when you save it if you have not appended the whole file, or if you have issued any W commands.

**Disk full**

ASM-86. There is not enough disk space for the output files (LST, H86, and SYM). You should either erase some unnecessary files or get another disk with more space and execute ASM-86 again.

ED. There is not enough disk space for the output file. This error can occur on the W, E, H, or X commands. If it occurs with X command you can repeat the command prefixing the filename with a different drive. Otherwise, you can try the recovery methods described above under the Directory full message.

**Disk read error - {filespec}**

ASM-86. A source or include file cannot be read properly. This is usually the result of an unexpected end-of-file. Correct the problem in your source file.

## CP/M-86 Messages (continued)

DDT-86. The disk file specified in an R command cannot be read properly. This is usually the result of an unexpected end-of-file. Correct the problem in your file.

GENCMD. The specified hex file cannot be read properly. This is usually the result of an unexpected end-of-file. Correct the problem by regenerating the H86 file.

PIP. The input disk file specified in a PIP command cannot be read properly. This is usually the result of an unexpected end-of-file. Correct the problem in your file.

**Disk write error - {filespec}**

DDT-86. A disk write operation cannot be successfully performed during a W command, probably due to a full disk. You should either erase some unnecessary files or get another disk with more space and execute ASM-86 again.

PIP. A disk write operation cannot be successfully performed during a PIP command, probably due to a full disk. You should either erase some unnecessary files or get another disk with more space and execute PIP again.

SUBMIT. The SUBMIT program cannot write the \$\$\$SUB file to the disk. Erase some files, or select a new disk and try again.

**Error on line nnn message**

SUBMIT. The SUBMIT program displays its messages in the format shown above, where nnn represents the line number of the SUBMIT file. Refer to the message following the line number

**ERROR: BAD PARAMETER**

PIP. An illegal parameter has been entered in a PIP command. Retype the entry correctly.

## CP/M-86 Messages (continued)

**ERROR: CLOSE FILE - {filespec}**

PIP. An output file cannot be closed. You should take appropriate action after checking to see if the correct disk is in the drive and that the disk is not write-protected.

**ERROR: DISK READ - {filespec}**

PIP. The input disk file specified in a PIP command cannot be read properly. This is usually the result of an unexpected end-of-file. Correct the problem in your file.

**ERROR: DISK WRITE - {filespec}**

PIP. A disk write operation cannot be successfully performed during a PIP command, probably due to a full disk. You should either erase some unnecessary files or get another disk with more space and execute PIP again.

**ERROR: FILE NOT FOUND - {filespec}**

PIP. An input file that you have specified does not exist.

**ERROR: HEX RECORD CHECKSUM - {filespec}**

PIP. A hex record checksum was encountered during the transfer of a hex file. The hex file with the checksum error should be corrected, probably by recreating the hex file.

**ERROR: INVALID DESTINATION**

PIP. The destination specified in your PIP command is illegal. You have probably specified an input device as a destination.

**ERROR: INVALID FORMAT**

PIP. The format of your PIP command is illegal. See the description of the PIP command.

## CP/M-86 Messages (continued)

**ERROR: INVALID HEX DIGIT - {filespec}**

GENCMD and PIP. An invalid hex digit has been encountered while reading a hex file. The hex file with the invalid hex digit should be corrected, probably by recreating the hex file.

**ERROR: INVALID SEPARATOR**

PIP. You have placed an invalid character for a separator between two input filenames.

**ERROR: INVALID SOURCE**

PIP. The source specified in your PIP command is illegal. You have probably specified an output device as a source.

**ERROR: INVALID USER NUMBER**

PIP. You have specified a User Number greater than 15. User Numbers are in the range 0 to 15.

**ERROR: NO DIRECTORY SPACE - {filespec}**

PIP. There is not enough directory space for the output file. You should either erase some unnecessary files or get another disk with more directory space and execute PIP again.

**ERROR: QUIT NOT FOUND**

PIP. The string argument to a Q parameter was not found in your input file.

**ERROR READING HELP.HLP INDEX**

HELP. The HELP.HLP file used by the HELP command is invalid. This can be caused by an unexpected end-of-file. The distributed HELP.HLP file should be copied to the disk from the CP/M-86 system disk.

## CP/M-86 Messages (continued)

**ERROR: START NOT FOUND**

PIP. The string argument to an S parameter cannot be found in the source file.

**ERROR: UNEXPECTED END OF HEX FILE - {filespec}**

PIP. An end-of file was encountered prior to a termination hex record. The hex file without a termination record should be corrected, probably by recreating the hex file.

**ERROR: USER ABORTED**

PIP. You aborted a PIP operation by pressing a key.

**ERROR: VERIFY - {filespec}**

PIP. When copying with the V option, PIP found a difference when rereading the data just written and comparing it to the data in its memory buffer. Usually this indicates a failure of either the destination disk or drive.

**\*\* ERROR NO: 0 \*\* Illegal first item**

ASM-86. The first item on a source line is not a valid identifier, directive, or mnemonic. Example:

```
1234H
```

**\*\* ERROR NO: 1 \*\* Missing pseudo instruction**

ASM-86. The first item on a source line is a valid identifier, and the second item is not a valid directive that can be preceded by an identifier. Example:

```
THIS IS A MISTAKE
```

**\*\* ERROR NO: 2 \*\* Illegal pseudo instruction**

ASM-86. Either a required identifier in front of a pseudo instruction is missing, or an identifier appears before a pseudo instruction that does not allow an

## CP/M-86 Messages (continued)

**\*\* ERROR NO: 3 \*\* Double defined variable**

ASM-86. An identifier used as the name of a variable is used elsewhere in the program as the name of a variable or label. Example:

```
X      DB      5
      . . .
X      DB      123H
```

**\*\* ERROR NO: 4 \*\* Double defined label**

ASM-86. An identifier used as a label is used elsewhere in the program as a label or variable name. Example:

```
LAB3: MOV      BX,5
      . . .
LAB3: CALL     MOVE
```

**\*\* ERROR NO: 5 \*\* Undefined instruction**

ASM-86. The item following a label on a source line is not a valid instruction. Example:

```
DONE: BAD      INSTR
```

**\*\* ERROR NO: 6 \*\* Garbage at end of line - ignored**

ASM-86. Additional items were encountered on a line when ASM-86 was expecting an end-of-line. Examples:

```
NOLIST 4
MOV     AX,4   RET
```

**\*\* ERROR NO: 7 \*\* Operand(s) mismatch instruction**

ASM-86. Either an instruction has the wrong number of operands, or the types of the operands do not match. Examples:

```
X      MOV     CX,1,2
      DB      0
      MOV     AX,X
```

## CP/M-86 Messages (continued)

**\*\* ERROR NO: 8 \*\* Illegal instruction operands**

An instruction operand is improperly formed. Examples:

```
MOV [BP+SP],1234
CALL BX|1
```

**\*\* ERROR NO: 9 \*\* Missing instruction**

ASM-86. A prefix on a source line is not followed by an instruction. Example:

```
REPZ
```

**\*\* ERROR NO: 10 \*\* Undefined element of expression**

ASM-86. An identifier used as an operand is not defined or has been illegally forward referenced. Examples:

```
      JMP    X
A     EQU   B
B     EQU   5
      MOV   AL,B
```

**\*\* ERROR NO: 11 \*\* Illegal pseudo operand**

ASM-86. The operand in a directive is invalid. Examples:

```
      X     EQU   OAGH
      TITLE UNQUOTED STRING
```

**\*\* ERROR NO: 12 \*\* Nested "IF" illegal - "IF" ignored**

ASM-86. The maximum nesting level for IF statements has been exceeded.

**\*\* ERROR NO: 13 \*\* Illegal "IF" operand - "IF" ignored**

ASM-86. Either the expression in an IF statement is not numeric, or it contains a forward reference.

## CP/M-86 Messages (continued)

**\*\* ERROR NO: 14 \*\* No matching "IF" for "ENDIF"**

ASM-86. An ENDIF statement was encountered without a matching IF statement.

**\*\* ERROR NO: 15 \*\* Symbol illegally forward referenced - neglected**

ASM-86. The indicated symbol was illegally forward referenced in an ORG, RS, EQU, or IF statement.

**\*\* ERROR NO: 16 \*\* Double defined symbol - treated as undefined**

ASM-86. The identifier used as the name of an EQU directive is used as a name elsewhere in the program.

**\*\* ERROR NO: 17 \*\* Instruction not in code segment**

ASM-86. An instruction appears in a segment other than a CSEG.

**\*\* ERROR NO: 18 \*\* File name syntax error**

ASM-86. The filespec in an INCLUDE directive is improperly formed. Example:

```
INCLUDE FILE.A86X
```

**\*\* ERROR NO: 19 \*\* Nested INCLUDE not allowed**

ASM-86. An INCLUDE directive was encountered within a file already being included.

**\*\* ERROR NO 20 \*\* Illegal expression element**

ASM-86. An expression is improperly formed. Examples:

```
      X     DB     12X
      DW     ( 4 * )
```

## CP/M-86 Messages (continued)

**\*\* ERROR NO: 21 \*\* Missing type information in operand(s)**

ASM-86. Neither instruction operand contains sufficient type information. Example:

```
MOV [BX],10
```

**\*\* ERROR NO: 22 \*\* Label out of range**

ASM-86. The label referred to in a call, jump, or loop instruction is out of range. The label can be defined in a segment other than the segment containing the instruction. In the case of short instructions (JMPS, conditional jumps and loops), the label is more than 128 bytes from the location of the following instruction.

**\*\* ERROR NO: 23 \*\* Missing segment information in operand**

ASM-86. The operand in a CALLF or JMPF instruction (or an expression in a DD directive) does not contain segment information. The required segment information can be supplied by including a numeric field in the segment directive as shown:

```

CSEG 1000H
X:
    . . .
    JMPF X
    DD X

```

**\*\* ERROR NO: 24 \*\* Error in codemacro building**

ASM-86. Either a codemacro contains invalid statements, or a codemacro directive was encountered outside a codemacro.

**File exists**

You have asked CP/M-86 to create a new file using a file specification that is already assigned to another file. Either delete the existing file or use another file specification.

## CP/M-86 Messages (continued)

REN. You have tried to rename a file to a name already assigned to another file. Either delete the existing file or rename the file to a different name.

**File is Read Only**

ED. A Read Only file cannot be edited with the ED command: ED filespec. Use the command ED inputfilespec outputfilespec instead.

**File not found**

CP/M-86 cannot find the specified file. Check that you have entered the correct drive specification or that you have the correct disk in the drive.

ED. ED cannot find the specified file. Check that you have entered the correct drive specification or that you have the correct disk in the drive.

STAT. STAT cannot find the specified file. Check to see if the correct disk is in the drive.

**Filename required**

ED. You typed the ED command without a filename. Reenter the ED command followed by the name of the file you want to edit or create.

**HELP.HLP read error**

HELP. An error occurred in reading the HELP.HLP file. Usually this error is caused by an unexpected end-of-file. A new copy of the HELP.HLP file should be copied from the CP/M-86 system disk.

**Illegal disk drive**

COPYDISK. An invalid or nonexistent drive was specified.

## CP/M-86 Messages (continued)

**Insufficient memory**

DDT-86. There is not enough memory to load the file specified in an R or E command.

**Insufficient memory available for copy**

COPYDISK. There is not enough memory available to copy a track from the specified disk.

**Insufficient memory to create CMD file**

GENCMD. There is not enough memory to create a CMD file from the H86 file specified.

**Invalid Assignment**

STAT. An invalid drive or file assignment was attempted. This error message might be followed by a list of the valid file assignments that can follow a filename. If an invalid drive assignment was attempted the message Use: d:=RO is displayed showing the proper syntax for drive assignments.

**Invalid control character**

SUBMIT. The only valid control characters in the SUBMIT files of type SUB are ^A through ^Z. Note that in a SUBMIT file, the control character is represented by typing the circumflex, ^, not by pressing the Control key.

**Is this what you want to do (Y/N)?**

COPYDISK. If the displayed COPYDISK function is what you want performed, type Y.

**Memory not available**

CP/M-86. There is not enough memory available for loading the program specified.

## CP/M-86 Messages (continued)

**Memory request denied**

DDT-86. A request for memory during an R command cannot be fulfilled. Up to eight blocks of memory can be allocated at a given time.

**No file**

DIR, ERA, REN. CP/M-86 cannot find the specified file, or no files exist.

ASM-86. The indicated source or include file cannot be found on the indicated drive.

DDT-86. The file specified in an R or E command cannot be found on the disk.

**No .HLP file on the default drive**

HELP. The HELP.HLP file was not found on the default drive. It should be copied from the CP/M-86 system disk.

**No more directory space**

GENCMD. There is insufficient directory space for creating the output file. A new disk should be selected or unnecessary files erased.

**No space**

DDT-86. There is no space in the directory for the file being written by a W command.

**No SUB file present**

SUBMIT. For SUBMIT to operate properly, you must create a file with filetype of SUB. The SUB file contains normal CP/M-86 commands. Use one command per line.

## CP/M-86 Messages (continued)

**Output file exists, erase it**

ED. The destination filename already exists when you are placing the destination file on a different disk than the source. It should be erased or another disk selected to receive the output file.

**Parameter error**

ASM-86. A parameter in the command tail of the ASM-86 command was specified incorrectly. Example:

```
ASM86 TEST $$;
```

SUBMIT. Within the SUBMIT file of type SUB, valid parameters are \$0 through \$9.

**Permanent error on track n  
Permanent error, sector n**

COPYDISK. n is the track or sector number. A bad sector exists on the source disk if the error occurred during a track read. Otherwise, the bad sector is on the destination disk. If the destination disk has the error, reformat. If the error persists, discard the disk.

**Read error**

TYPE. An error occurred when reading the file specified in the TYPE command. Check the disk and try again. The STAT filespec command can diagnose trouble.

**Source and destination cannot be the same**

COPYDISK. The source and destination drives must be different, although drives A, B, C, and D can all be mapped to the same physical drive. The system prompts you to change disks when the drive changes.

## CP/M-86 Messages (continued)

**Source and destination disks must be the same type**

COPYDISK. Both the source and destination disks must have the same characteristics. (The STAT DSK: command displays the disk characteristics.)

**Symbol table overflow**

ASM-86. There is not enough memory for the symbol table. Either reduce the length and/or number of symbols, or reassemble on a system with more memory.

**Too many files****Too many entries in index table**

HELP. There is not enough memory available to run the HELP utility.

STAT. There is not enough memory for STAT to sort the files specified or more than 512 files were specified.

**Topic not found**

HELP. The topic requested does not exist in the HELP.HLP file. A topic should be selected from the menu displayed.

**Unable to find file HELP.HLP**

HELP. The HELP.HLP file cannot be found on the default drive. Copy it to the default drive from the CP/M-86 system disk.

**Use: [size] [ro] [rw] [sys] or [dir]**

STAT. This message results from an invalid set file attributes command. These are the only options valid in a STAT filespec [option] command.



## CP/M-86 Messages (continued)

Use: **STAT d:=RO**

STAT. An invalid STAT drive command was given. The only valid drive assignment in STAT is STAT d:=RO.

Verify error at s:o

DDT-86. The value placed in memory by a Fill, Set, Move, or Assemble command cannot be read back correctly, indicating bad user memory or an attempt to write to ROM or nonexistent memory at the indicated location.

## CP/M-86 BDOS FUNCTION SUMMARY

No.	Hex	Function Name	Input	Output
0*	00H	System Reset	DL=Abort Code	None
1	01H	Console Input	None	Char
2	02H	Console Output	DL=Char	None
3	03H	AXI: Input	None	AL=Char
4	04H	AXO: Output	DL=Char	None
5	05H	List Output	DL=Char	None
6*	06H	Direct Con I/O	DL=0FFH DL=Char	AL=Char Char/Stat None
7	07H	Get I/O Byte	None	AL=IOBYTE
8	08H	Set I/O Byte	DL=IOBYTE	None
9	09H	Print String	DX=Str Offset	None
10	0AH	Read Con Buffer	DX=Buf Offset	Chars
11	0BH	Get Con Status	None	AL=Con Stat
12	0CH	Return Vers No.	None	BX=Vers No.
13	0DH	Reset Disk Sys.	None	None
14	0EH	Select Disk	DL=Disk	None
15	0FH	Open File	DX=FCB Offset	AL=Rtn Code
16	10H	Close File	DX=FCB Offset	AL=Rtn Code
17	11H	Search For First	DX=FCB Offset	AL=Dir Code
18	12H	Search For Next	None	AL=Dir Code
19	13H	Delete File	DX=FCB Offset	AL=Rtn Code
20	14H	Read Sequential	DX=FCB Offset	AL=Rtn Code
21	15H	Write Sequential	DX=FCB Offset	AL=Rtn Code
22	16H	Make File	DX=FCB Offset	AL=Rtn Code
23	17H	Rename File	DX=FCB Offset	AL=Rtn Code
24	18H	Rtn Login Vector	BX=Login Vect	BX=Login Vect
25	19H	Rtn Current Disk	None	AL=Curr Dsk
26	1AH	Set DMA Address	DX=DMA Offset	None
27*	1BH	Get Addr (Alloc)	None	BX=Alloc Ofst ES=Seg Base
28	1CH	Write Protect Dsk	None	None
29	1DH	Get Read/Only Vec	None	BX=RO Vector
30	1EH	Set File Attrib	DX=FCB Offset	AL=Rtn Code
31*	1FH	Get Adr(Dsk Parm)	None	BX=DPB Ofst ES=Seg Base
32	20H	Set/Get User Code	DL=OFFH(get) DL=User # (set)	AL=Curr Code None
33	21H	Read Random	DX=FCB Offset	AL=Rtn Code
34	22H	Write Random	DX=FCB Offset	AL=Rtn Code
35	23H	Compute File Size	DX=FCB Offset	Random Rec Field Set
36	24H	Set Random Record	DX=FCB Offset	Random Rec Field Set
37*	25H	Reset Drive	DX=Drive Vect	AL=00H
40	28H	Write Ran/0 Fill	DX=FCB Offset	AL=Rtn Code
47	2FH	Chain to Program	DMA Buf:Cmd Ln	None
49	31H	Get Sysdat Addr.	None	BX=Sysdat Addr Ofst ES=Sysdat Addr Seg
50*	32H	Direct BIOS Call	DX=BIOS Dscptr	None
51*	33H	Set DMA Seg Base	DX=Base Addr	None
52*	34H	Get DMA Seg Base	None	BX=DMA Ofst ES=DMA Seg

## CP/M-86 BDOS Function Summary (continued)

No.	Hex	Function Name	Input	Output
53*	35H	Get Max Memory	DX=MCB Offset	AL=Rtn Code
54*	36H	Get Max Mem Abs	DX=MCB Offset	AL=Rtn Code
55*	37H	Allocate Memory	DX=MCB Offset	AL=Rtn Code
56*	38H	Alloc Abs Memory	DX=MCB Offset	AL=Rtn Code
57*	39H	Free Memory	DX=MCB Offset	None
58*	3AH	Free All Memory	None	None
59*	3BH	Program Load	DX=FCB Offset	AX=Rtn Code/ Base Pg Addr BX=Base Pg Addr

\* Function differs from or has been added to set of CP/M<sup>®</sup> 2.2 functions.

## CP/M-86 FILE CONTROL BLOCK

The File Control Block (FCB) data area is a sequence of 33 bytes for sequential access, or a sequence of 36 bytes if Random Access Functions 33, 34, 35, 36, or 40 are used. The default File Control Block normally located at offset 005CH from the DS register can be used for random access files because bytes 006DH, 007EH, 007FH are available for this purpose. Here is the FCB format, followed by definitions of each of its fields.

dr	fl	f2	/	/	f8	t1	t2	t3	ex	s1	s2	rc	d0	/	/	dn	cr	r0	r1	r2
----	----	----	---	---	----	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----

00 01 02 ... 08 09 10 11 12 13 14 15 16 ... 31 32 33 34 35

where:

- dr drive code (0 - 16)  
0 => use default drive for file  
1 => auto disk select drive A,  
2 => auto disk select drive B,  
...  
16=> auto disk select drive P.
- fl...f8 contain the file name in ASCII upper case, with high bit = 0
- t1,t2,t3 contain the file type in ASCII upper case, with high bit = 0  
t1', t2', and t3' denote the high bit of these positions,  
t1' = 1 => Read/Only file,  
t2' = 1 => SYS file, no DIR list
- ex contains the current extent number, normally set to 00 by the user, but in range 0 - 31 during file I/O
- s1 reserved for internal system use
- s2 reserved for internal system use, set to zero on call to OPEN, MAKE, SEARCH
- rc record count for extent "ex," takes on values from 0 - 128
- d0...dn filled in by CP/M, reserved for system use
- cr current record to read or write in a sequential file operation, normally set to zero by user
- r0,r1,r2 optional random record number in the range 0-65535, with overflow to r2, r0,r1 constitute a 16-bit value with low byte r0, and high byte r1

Both CP/M Version 2 and CP/M-86 perform directory operations in a reserved area of memory that does not affect write buffer content. Write buffer content is only affected in the case of Search and Search Next, where the directory record is copied to the current DMA address.

## CP/M-86 BASE PAGE VALUES

---

The CP/M-86 base page contains default values and locations initialized by the CCP and used by the transient program. The base page occupies the regions from offset 0000H through 00FFH relative to the DS register. The values in the base page for CP/M-86 include those of CP/M-80<sup>U,M</sup>, and appear in the same relative positions, as shown in the following figure.

DS + 0000:	LC0	LC1	LC2
DS + 0003:	BC0	BC1	M80
DS + 0006:	LD0	LD1	LD2
DS + 0009:	BD0	BD1	xxx
DS + 000C:	LE0	LE1	LE2
DS + 000F:	BE0	BE1	xxx
DS + 0012:	LS0	LS1	LS2
DS + 0015:	BS0	BS1	xxx
DS + 0018:	LX0	LX1	LX2
DS + 001B:	BX0	BX1	xxx
DS + 001E:	LX0	LX1	LX2
DS + 0021:	BX0	BX1	xxx
DS + 0024:	LX0	LX1	LX2
DS + 0027:	BX0	BX1	xxx
DS + 002A:	LX0	LX1	LX2
DS + 002D:	BX0	BX1	xxx
DS + 0030:	Not		
DS + 0033:	Currently		
DS + 0036:	Used		
DS + 005C:	Default FCB		
DS + 0080:	Default Buffer		
DS + 0100:	Begin User Data		

Each byte is indexed by 0, 1, and 2, corresponding to the standard Intel storage convention of low-, middle-, and high-order (most significant) byte. Unused bytes are marked by xxx. LC is the length of the code group location (24 bits, where the 4 high-order bits equal zero).