DIGITAL RESEARCH®

CP/M® 2.2

QUICK REFERENCE CARD

A ZENITH SOFTWARE PRODUCT

HEATH | ZENITH data systems

597-2315-1

Conventions

ufn unambiguous file name uniquely specifies a single file. It consists of a primary name of up to 8 characters and (optionally) an extension of up to 3 characters separated by a period. The file name can not contain any of the following special characters:

< > . , ; : = ? * [ ]

The extension is commonly used to describe the type of the file. Examples:

.ASM	assembly language source file
.HEX	Intel hex format
.COM	executable "command" file
.SUB	submit file
.BAS	BASIC file
.DAT	misc. data file

afn ambiguous file name - similar to the unambiguous file name, except it can also use "wild cards" to match a set of file names. A "*" is used to match an entire primary name or extension, a "?" matches a single alphanumeric character

[ ] text enclosed in square brackets is used as "switches" to enable or disable options, the square brackets are typed by the user

: prompts ending in : require only single character input — no carriage return is required

all other input lines should be terminated with a RETURN

↑ preceeding a character means to depress the CTRL key and while holding it down, type the indicated character

{ } curved brackets are used to indicate optional input, the brackets are not typed by the user

BACKSPACE deletes preceeding character, and backspaces video terminal cursor

DELETE deletes preceeding character, and echos the deleted character, useful for printing terminals

↑C warm boot (when first character of line)

↑E physical end of line

↑J (line feed) terminates current input

↑M (carriage return) terminates current input

↑R retype input line to present position

↑U delete entire input line

↑X backspace to beginning of current line

↑P toggle printer on/off, printer duplicates console output when on

CCP Commands

DIR afn	displays list of all files matching afn DIR is equivalent to DIR *.*
ERA afn	erases all files matching afn
TYPE ufn	type the contents of the file on console
USER n	change active user number to n (0 - 15)
SAVE n ufn	save n pages (256 bytes/page) starting at 0100H (TPA) as file ufn
REN ufn 1=ufn2	rename ufn2 to be ufn1 NEWNAME = OLDNAME

ASM - Assembler

Command line format

ASM FILE.MNO where
M = source drive letter
N = drive for hex output file
or Z for no hex file
O = drive for list file
or X to send listing to console
or Z for no listing file

Error messages

D	data error
E	expression - ill-formed or can not be computed at assembly time
L	label error - illegal position
N	not implemented in this version of ASM
O	overly complex expression
P	phase error - label has different values on 2 passes through program
R	register error - specified register illegal in this context
V	value error - ill-formed operand

DDT - Dynamic debugging tool

(all values in hex)

Command Line: **DDT cr**
or **DDT file.ext cr**

Axxxx	line by line assembler starting at xxxxH
Dxxxx{,yyyy}	dump memory in hex and ASCII
Fxxxx,yyyy,bb	fill memory from xxxx through yyyy with hex byte bb
G{xxxx}{,yyyy}	go starting at PC or xxxx if supplied, optional breakpoint at yyyy
Hxxxx,yyyy	hexadecimal sum, difference
Itext	use text to set up standard file control block
L{xxxx}{,yyyy}	disassemble starting at xxxx
Mxxxx,yyyy,zzzz	move block from xxxx through yyyy to zzzz
R{xxxx}	read file with optional offset
Sxxxx	substitute starting at xxxx (terminate with illegal hex character)
T{n}	trace n steps (default = 1)
U{n}	same as T, but intermediate steps are not displayed
X{regpair}	examine CPU registers, if register pair is specified, permit altering it.

ED - Editor

Command Line: **ED y:file.ext** {x:}

edits file.ext, if it does not exist it will be created. At the end of an edit session, the original file will be renamed file.BAK, the new file will be called file.ext. If specifed, x: specifies the drive for the output file (useful for editing files that are ğreater than half the disk).

most of the editor commands operate on text beginning at an imaginary character pointer, **cp** which can be moved throughout the memory buffer

nA append the next n lines from the source file to the end of the memory buffer, # A gets all, OA half fills buffer

±B move **cp** to beginning, −B to end

±nC move **cp** n characters

±nD delete n characters

E end edit, output file, rename original .BAK

nFstring(cr or ↑Z) find nth occurrence of string

H end edit, close, then reopen files

I insert characters at **cp**

ICR accepts input lines until a **↑Z** is typed an abbreviated form is **Itext . . text(↑Z or cr)** which inserts the specified text on the same line as the **cp**, then positions **cp** immediately following the text if **↑Z** is used or at the beginning of the next line if **cr** is used

±nK kill n lines

±nL move **cp** n lines

nM define macro command sequence

nN like **F**, but automatically brings more of the source file into memory, if not in current buffer

O return to original file, restart edit

±nP move and print n pages

Q quit, making no changes

R read library file

nSstring1↑Zstring2↑Z substitute string2 for string1 n times

±nT type n lines

±U translate to upper case if +, disable translation if −

±V + turn on line numbers, − turn off line numbers, OV prints free/total memory buffer size

nW write first n lines to file, W writes all, OW writes lines until the buffer is half empty

nX copy n lines to temporary X$$$$$$$.LIB file, OX empties file

±n CR equivalent to **±nLT**

when using line numbers, nnn: preceeding a command means to start the execution at line nnn. :mmm means the final line is mmm. nnn::mmm specifies a range of line numbers for a command

if # used in place of an integer n, the largest value of n allowed (65535) is used.

LIST utility

```
LIST ufn { . . .{switches}
or
LIST cr
*  ufn {ufn  . . . } {switches}
*  ufn . . .
*  cr
```

where the switches are:

[D xxxxxxxxxx] the first ten characters (or to the end of the field) are used as a date and printed just to the left of the page number in heading

[H text] the first sixty characters (or to the end of the field are used as header and printed on the top line of each page. If not present, defaults to file name.

[L nn] sets number of lines to user text per page, default = 60, L = 00 turns off pagination

[N] no heading line (which implies no page numbering)

[T nn] tab stops every nn positions, default= 8

[P nn] start page numbering with nn, default= 1

[U] force all output to upper case

[C nn] number of copies of output created by this command line, default = 1

[E] erases files on that command line after printing them

When used in the prompt mode, all settings (with the exceptions of the starting page number and erasure) remain in effect until the program is terminated.

Output can be aborted by typing any character on the console keyboard during printing.

SUBMIT - utility

SUBMIT ufn {parameters} cr

The SUBMIT command allows a series of CP/M commands to be executed automatically. The commands are listed sequentially in the file "ufn". The file must have a file type of ".SUB" and reside on drive A. SUBMIT allows parameters to be substituted into the commands of the SUB file. These parameters are typed in the SUBMIT command line after the file name. If parameters are to be substituted, the SUB file will contain "$" parameters of the form

$1 $2 $3 ... $n

interspaced through the file. These parameters correspond to the actual number of parameters which will be included when the file is submitted for execution. As the file is executed, the $1 will be replaced with the first parameter listed in the command line. Similarly the $2 will be substituted with the second parameter etc.

XSUB - SUBMIT extender

When the XSUB command is included as the first line of the submit file, the power of the SUBMIT utility is extended to accept line input to programs as well as normal CP/M commands.

PIP - peripheral interchange program

purpose: file transfer utility

```
PIP command line cr
or
PIP cr
*command line cr
*command line cr
*cr
```

the command line is of the form:

destination = **source1** {, **source2** . . . . } {**switches**}

where the destination is the file or device to receive data and the sources are a series of one or more files or devices that provide the data

In addition to the standard logical devices, PIP may reference:

NUL:	source of 40 nulls
EOF:	source of a CP/M end-of-file (↑Z)
INP:	special PIP input source that can be patched into PIP
OUT:	special PIP output destination that can be patched into PIP
PRN:	same as LST: except tabs are expanded to every 8th column, lines are numbered, and page ejects are inserted every 60 lines

Switches

switches are enclosed in square brackets ([ ])

B	block mode transfer, data buffered until a ctl-S received from source
Dn	delete characters that extend past column n
E	echo transfers to console as performed
F	remove form feeds from file
Gn	get file from user n (0 - 15)
H	hex data transfer - all data checked for proper Intel hex file format, non-essential characters are removed
I	ignore :00 records in Intel hex file, automatically sets "H"
L	translate upper case to lower case
N	add line numbers to each line, starting at 1, incrementing by 1. Leading zeros suppressed, followed by a colon. N2 includes leading zeroes, and follows number with a tab
O	object file transfer, embedded CP/M end-of-file marks (↑Z) are ignored
Pn	include page ejects every n lines
Qs↑z	quit copying from source when string s is encountered
R	read system files (those with $SYS set)
Ss↑z	start copying from source when string s is encountered
Tn	expand tabs to every nth column
U	translate lower case to upper case
V	verify that data has been copied correctly by rereading after the write operation (destination must be a disk file)
W	write over R/O files without console interrogation
Z	zero the parity bit of each ASCII character

STAT - system status utility

STAT cr print the amount of free space on all active drives in K (1024 bytes), also indicate the read/write status of the drive

STAT x: cr selects specified disk, then displays amount of free space in K

STAT afn cr display size, access privileges, and file names (in alphabetical order) of all files matching afn. file names in "()" mean the file is flagged $SYS

STAT X:=R/O cr flags drive x read only (until next warm boot)

STAT VAL: cr displays possible STAT commands, and permissible logical to physical device mappings

STAT DEV: cr display the current logical to physical device mapping

STAT afn indicator cr

set the permanent file indicator for afn where the permissible indicators are:

$R/O	read-only
$R/W	read/write
$SYS	system file - hidden from DIR
$DIR	show file in directory

STAT DSK: list the drive characteristics for all currently active drives

STAT x: DSK: list the drive characteristics for the specified drive

STAT USR: display active user number and all user numbers that have files on the current default disk

BDOS Call Summary

#	Name	Input	Output
0	System Reset	none	none
1	Console Input	none	A = char
2	Console Output	E = char	none
3	Reader Input	none	A = char
4	Punch Output	E = char	none
5	List Output	E = char	none
6	Direct Console I/O	see manual	see manual
7	Get IOBYTE	none	A = IOBYTE
8	Set IOBYTE	E = IOBYTE	none
9	Print String Buffer	DE -> buffer	none
10	Read Console Buffer	DE -> buffer	see manual
11	Get Console Status	none	A = 00 / FF
12	Get Version Number	none	HL = version
13	Reset Disk System	none	see manual
14	Select Disk	E = Disk #	see manual
15	Open File	DE -> FCB	A = dir code
16	Close File	DE -> FCB	A = dir code
17	Search for First	DE -> FCB	A = dir code
18	Search for Next	none	A = dir code
19	Delete File	DE -> FCB	A = dir code
20	Read Sequential	DE -> FCB	A = err code
21	Write Sequential	DE -> FCB	A = err code
22	Make File	DE -> FCB	A = dir code
23	Rename File	DE -> FCB	A = dir code
24	Get Login Vector	none	HL = login
25	Get Current Disk	none	A = disk #
26	Set DMA address	DE = adr	none
27	Get Allocation ADR	none	HL -> Alloc
28	Write Protect Disk	none	see manual
29	Get R/O Vector	none	HL = R/O Vec
30	Set File Attributes	DE -> FCB	see manual
31	Get Disk Params Adr	none	HL -> DPB
32	Set/Get User Code	see manual	see manual
33	Read Random	DE -> FCB	A = err code
34	Write Random	DE -> FCB	A = err code
35	Compute File Size	DE -> FCB	r0, r1, r2
36	Set Random Record	DE -> FCB	r0, r1, r2
37	Reset Drive	DE = drive vec	0
40	Write Rand w/0 fill	DE -> FCB	A = err code

FCB Format

default FCB begins at 005CH

byte offset from beginning (hex)

00 drive (0=default, 1=A, 2=B, . .)
01-08 upper case file name, left justified, blank filled
09-0B upper case extension
if msb of byte 09 = 1 then R/O file
if msb of byte 0A = 1 then SYS file
OC extent, normally set to 0 by user
OD-OE reserved for system use
OF record count for current extent
10-1F reserved for system use
20 current record to read or write in a sequential operation, normally initially zeroed by user
21 r0 r0, r1, r2 form random record number 0 - 65535
22 r1 (with overflow to r2), r0 is the least
23 r2 significant byte