QX-10

COMMAND SUMMARY

QX-10 MultiFonts CP/M

Built-in commands

DIR
Lists the file directory of the disk in the currently logged in drive.

DIR d
Lists the file directory of the disk in drive d.

DIR d:filename.typ
Lists the name of the specified file if it exists on the specified disk.
(Wildcard characters * and ? can be used in filename.typ.)

ERA d:filename.typ
Erases the specified file on the disk in drive d. (Wildcard characters * and ? can be used in filename.typ.)

REN d:newname.typ = oldname.typ
Changes the name of the file on drive d from oldname to newname.

SAVE n d:filename.type
Saves n 256-byte pages of memory starting at 100H to the specified disk.

TYPE d:filename.typ
Lists the contents of the specified file on the disk in drive d.

USER n
Changes the user area to n.

d:
Changes the logged-in drive to d.

Transient commands

ASM d:filename
Assembles the specified ASM file on the disk in drive d.

ASM filename.shp
Assembles the specified ASM file on the disk in drive s,

and outputs the HEX type object file to drive h and the PRN type list file to drive p.

DDT
Activates DDT.

DDT filename.type
Activates DDT and loads the specified file.

DUMP d:filename.type
Lists the contents of the specified file to the display in hexadecimal.

ED d:filename.typ
Activates the editor to generate or edit the specified file.

LOAD d:filename
Generates a COM file from the specified HEX file on the disk in drive d.

MOVCPM n *
Reconfigures the CP/M system for nK bytes of memory.

MOVCPM * *
Reconfigures the CP/M system for available memory.

PIP
Activates the PIP command.

PIP d: = s:filename.type
Copies the specified file from drive s to drive d. (Wildcard characters * and ? can be used in filename.typ.)

PIP LST: = filename.typ
PUN:
CON:
Transfers the contents of the specified file to the specified device.

PIP filename.typ = RDR:
Transfers data from RDR: to the specified file.

PIP d:newname.typ = s1:name1.typ,s2:name2.type,...
Concatenates the specified files to generate a new file.

STAT
Displays the amount of free space on the disk in the currently logged-in drive and other drives which have been at least once used, as well as their R/W attribute.

STAT dr:
Displays the amount of free space on the disk in the specified drive.

STAT dr:filename.typ
Displays the size and attributes of the specified file.

STAT dr:filename.typ $S
Displays the size and attributes of the specified file in detail.

STAT dr:filename.typ $R/O
Sets the specified file to read only.

STAT dr:filename.typ $R/W
Makes it possible to read or write the specified file.

STAT dr:filename.typ $SYS
Sets the SYS attribute of the specified file.

STAT dr:filename.typ $DIR
Sets the DIR attribute of the specified file.

STAT DEV:
Displays the current physical-to-logical device assignments (that is, the contents of IOBYTE).

STAT VAL:
Displays instructions for specifying the operand of the STAT command.

STAT DSK:
Displays the status of the currently logged-in drive and other drives which have been used at least once.

STAT dr:DSK:
Displays the status of the specified disk.

STAT USR:
Displays the current user number and user numbers which have active files on the current disk.

STAT dr: = R/O
Sets the specified drive to read only. The read only state remains effective until a cold or warm start is made.

STAT logical: = physical:
Assigns the specified physical device to the specified logical device.

SUBMIT filename parameters
Executes the commands in command procedure file "filename.SUB" using parameters.

SYSGEN
Copies the system from the system disk to a disk or memory.

XSUB
Extends the function of SUBMIT.

Transient commands unique to QX-10

AUTOST
Automatically starts the specified program upon a cold start.

CHARADEF
Allows you to design dot matrix patterns (which can be displayed or printed by application programs), and store them on a system disk.

CONFIG
Makes it possible to reconfigure the system for different printers and RS-232C communication formats, to set the clock, and to select one of the 8 international character sets.

DIRINIT
Erases and initializes the disk's file directory.

DISKCOPY
Formats a disk, then makes a copy of another disk and verifies the copy; copies a disk and verifies the copy; or compares the contents of two disks.

FORMAT
Formats a new disk and writes ID information on the

disk after formatting it.

MFBASIC
Activates the MFBASIC.

MFONT
Sets CP/M in the MultiFont (MF) mode.

NORM
Sets CP/M in the normal (non-MF) mode.

PFKSET
Assigns strings to the programmable function keys from the system's PFK table.

PFKSET /L
Allows you to change the contents of the system's PFK table.

PFKSET x:ufn
Assigns strings to the programmable function keys from file ufn in drive x. PFK is assumed as the file type if no file type is specified.

PFKSET x:ufn /L
Allows you to change the contents of your own PFK table or generate a new one. If the filetype is omitted, PFK is assumed.

TERM
Makes it possible to use the QX-10 either as a terminal or host computer for remote operation. It also allows specification of various communications conditions, such as bit rate, parity bit, stop bits and word length.

Control Key commands

CTRL and C : Performs warm boot of CP/M.

CTRL and E : Physically terminates a line.

CTRL and H : Performs the same function as the "BS" key.

CTRL and I : Performs the same function as the "TAB" key.

CTRL and P : Entering this command once causes all information displayed on the CRT to be printed on the printer (LST:). Entering it again restores normal operation.

CTRL and R : Redisplays the current command line.

CTRL and S : Entering this command once suspends any operation. Entering it again restarts operation.

CTRL and U : Invalidates the current entry and moves the cursor to the top of the next line.

CTRL and X : Deletes the current entry and moves the cursor to the top of the current line.

DDT subcommands

lufn

Catalogs file name ufn in the default file control block at address 5CH.

R or Roffset

Load the file specified in the default file control block into memory starting at 100H (or 100H + offset).

Hnnnn mmmm

Calculates the results of nnnnH + mmmmH and nnnnH-mmmmH.

D, Daddress or Daddress1 address2

This command displays the contents of the main memory area starting at "last display address + 1" in both hexadecimal and ASCII form.

Faddress1 address2 c

Fills the memory area from addressl to address2 with the hexadecimal constant c.

Maddress1 address2 address3

Transfers the contents of the memory area from address1 to address2 to the area starting at address3.

Saddress

Used to change the contents of memory.

Aaddress

Assembles 8080 mnemonic instructions entered following the indicated addresses.

L, Laddress or Laddress1 address2

Lists the contents of the specified memory area in 8080 mnemonics.

X or Xr

Displays the registers contents and flag settings, and allows you to change their contents.

Gaddr,bp1 bp2

Starts execution of a program loaded by the DDT command. Execution starts at the address specified in the subcommand, and up to two break points can be set. Tm traces m steps of the program starting at the program counter address.

Um

Traces m steps of the program starting at the program counter address. Trace results are displayed only for the last instruction executed.

ED subcommands

nA

Appends the n lines of text from the disk to the contents of the edit buffer.

0A

Appends text from the disk to the contents of the edit buffer until half of the edit buffer is filled.

±B

Moves the character pointer (CP) to the beginning (+) or end (-) of the text in the edit buffer.

±nL

Moves the CP ±n lines from the current line and locates it at the beginning of the new line.

±n

Moves the CP ±n lines from the current line, locates it at the beginning of the new line, and displays that line. (= ±nLT)

0

Moves the CP to the beginning of the current line and displays the current line. (=0LT)

[←┘]

Moves the CP to the beginning of the next line and displays that line. (Same as with LT.)

n:

Moves the CP to the beginning of line n.

±nC

Moves the CP ±n characters from its current position.

±nT

Displays the ±n lines from the current CP position.

0T

Displays the current line from its beginning to the CP position.

n::mT

Displays the text from line n to line m.

±np

Displays the ±n pages from the current CP position. (1 page = 23 lines)

nFstring

Locates the n'th occurence of the character string following the current CP position and locates the CP to the end of that string.

nNstring

Appends the remainder of the text from the disk and performs the same function as nFstring.

±nD

Deletes the n characters preceding (-) or following (+) the CP position.

±nK

Deletes the n lines preceding (-) or following (+) the line on which the CP is located. When + is specified, the current line is deleted; otherwise, it is excluded.

I

Enters the insert mode to allow insertions to be made.

Istring^Z

Inserts the specified string at the current CP position.

Istring [←┘]

Inserts the specified string and a CR/LF in the current CP position to generate a new line.

R

Inserts lines previously saved as temporary file X$$$$$$$.LIB (with the nx subcommand) at the current CP position.

Rfilename

Inserts a file of type LIB at the current CP position. (The file must exist on the disk which contains the original source file or on which the new file is to be stored.

nSstring1^Zstring2

Replaces the n'th occurence of string1 following the current CP position with string2, then locates the CP at the end of string2.

nJstring1^Zstring2^Zstring3

Inserts string2 following the n'th occurence of string1 following the current CP position, then deletes all characters from the end of string2 to the beginning of string3. string3 is not deleted.

nX

Appends the n lines of the text following the CP to the contents of temporary file X$$$$$$$.LIB.

R

Inserts the contents of temporary file X$$$$$$$.LIB at the current CP position.

0X
Empties temporary file X$$$$$$$.LIB.

±U
+U causes all characters to be converted to upper case as they are entered; -U resets this function.

nMsubcommand
Repeats the specified subcommand n times.

Msubcommand
Repeats the specified subcommand until the end of buffer is reached. (=0M, 1M)

±V
-V makes line numbers invisible. +V resets this mode.

0V
Displays the amount of free space and total space in the edit buffer.

nZ
Delays display by approximately n/4 seconds.

nW
Saves the n lines from the beginning of the edit buffer to temporary file filename.$$$. Lines saved are deleted from the edit buffer.

E
Saves the contents of the edit buffer and the remainder of the source file to the disk and terminates the ED command.

H
Saves the contents of the edit buffer and the remainder of the source file to the disk, deletes the contents of the edit buffer, and restarts the ED command for the same file name.

O
Cancels and clears all preceding editing operations to allow editing to be started over.

Q
Terminates the ED command; no change is made on the original file.

n:subcommand
Moves the CP to the beginning of the current line and executes subcommand.

:nsubcommand
Executes subcommand for the text between the CP and line n.

Notes:
* the CP is an abbreviation for character pointer.
* The + symbol can be omitted.
* n can be omitted when it is 1.
* When # is specified as n for subcommands A and W, all lines of the text are indicated.
* More than one command can be entered at the same time.

PIP parameters

[B] (Block)
When specified, PIP performs block mode transfer.

[Dn] (Delete)
When specified, PIP deletes characters exceeding column n of each line.

[E] (Echo)
When specified, data transferred is also output to the console.

[F] (Form feed)
When specified, form feed characters (0CH) are removed from data transferred.

[Gn]
When specified, data can be transferred from another user area.

[H] (Hex format)
When specified, PIP checks to confirm that data transferred is in Intel HEX format. (If not, operation is terminated.)

[I] (Ignore null)
When specified, PIP ignores null records (00:) and checks to confirm that data transferred is Intel HEX format. (If not, operation is terminated.)

[L] (Lower case)
When specified, all upper case characters are converted to lower case characters.

[N] (Line number)
When specified, line numbers are added to the beginning of each line. Specifying "N2" adds zeros to the beginning of each line number.

[O] (Object files)
When specified, PIP ignores the physical end of file code (1AH) during concatenation and transfer. This makes it possible to transfer files other than ASCII files.

[Pn] (Page eject)
When specified, PIP makes a page eject every n lines. When n is 1 or omitted, 60 is assumed.

[Qstring^Z] (Quit)
When specified, PIP quits transfer when the specified string is detected.

[Sstring^Z] (Start)
When specified, PIP searches the data to be transferred for the specified string, then starts transfer from the point at which it is detected.

[R] (Read)
Makes it possible to transfer .SYS files. (The [W] parameter is set automatically when [R] is specified.)

[Tn] (Tab)
When specified, the tab width for transfer is set to n columns.

[U] (Upper case)
When specified, all lower case characters are converted to upper case characters.

[V] (Verify)
When specified, each file is verified as it is copied.

[W] (Write in R/O)
When specified, the R/O attributes of destination files are ignored.

[Z] (Zero parity)
When specified, parity bits of data received are turned to zero.

BDOS function call

Entry point: 0005H

No.	Register C	Function	Entry parameter	Return parameter
0	(00H)	System reset	None	None
1	(01H)	Console input	None	A: input character
2	(02H)	Console output	E: character to be output	None
3	(03H)	Reader input	None	A: input character
4	(04H)	Punch output	E: character to be output	None
5	(05H)	List output	E: character to be output	None
6	(06H)	Direct console I/O	E: FFH (input) E: character (output)	A: input character or 00H (no input)
7	(07H)	Get I/O byte	None	A: IOBYTE
8	(08H)	Set I/O byte	E: IOBYTE	None
9	(09H)	Print string	DE: string buffer address	None
10	(0AH)	Read console buffer	DE: read buffer address	Buffer: input string
11	(0BH)	Get console status	None	A: FFH (ready) 00H (not ready)
12	(0CH)	Return version No.	None	H: CPM or MPM L: version number

No.	Register C	Function	Entry parameter	Return parameter
13	(0DH)	Reset disk system	None	None
14	(0EH)	Select disk	E: drive number	None
15	(0FH)	Open file	DE: FCB address	FBC: directory information A: directory code A: FFH (no file)
16	(010H)	Close file		
17	(11H)	Search for first file		
18	(12H)	Search for next file	None	A: directory code A: FFH (no file)
19	(13H)	Delete file	DE: FCB address	
20	(14H)	Read sequential		A: 00H (normal completion)
21	(15H)	Write sequential		A: 00H (normal completion)
22	(16H)	Make file		A: directory code A: FFH (directory full)
23	(17H)	Rename file		A: directory code A: FFH (no file)
24	(18H)	Return login vertor	None	HL: login vector
25	(19H)	Return current disk No.	None	A: drive number

No.	Register C	Function	Entry parameter	Return parameter
26	(1AH)	Set DMA address	DE: DMA address	None
27	(1BH)	Get address (ALLOC)	None	HL: allocation vector address
28	(1CH)	Write protect disk	None	None
29	(1DH)	Get R/O vector	None	HL: R/O vector
30	(1EH)	Set file attributes	DE: FCB address	A: directory code
				A: FFH (no file)
31	(1FH)	Get address (DPB)	None	HL: DPB base address
32	(20H)	Set/get user code	E: FFH (get)	A: current user code
			E: User code (Set)	
33	(21H)	Read random	DE: FCB address	A: 00H (normal completion)
34	(22H)	Write random		A: error code (error)
35	(23H)	Compute file size		R0, R1 and R2 of FCB: file size
36	(24H)	Set random record number		R0, R1 and R2 of FCB: random record number
37	(25H)	Reset drive	DE: drive vector	A: 00H
40	(28H)	Write random with zero file	DE: FCB address	A: 00H (normal completion)
				A: error code (error)

Relationship between physical devices and logical devices (IOBYTE)

Logical device / Bit position	LST: 7 6	PUN: 5 4	RDR: 3 2	CON: 1 0
Physical device	0 0 TTY: (none)	0 0 TTY: (printer)	0 0 TTY: (keyboard)	0 0 TTY: (keyboard(O)) (keyboard(I))
	0 1 CRT: (CRT)	0 1 PTP: (none)	0 1 PTR: (none)	0 1 CRT: (CRT(O)) (keyboard(I))
	1 0 LPT: (printer)	1 0 UP1: (RS232C)	1 0 UR1: (RS232C)	1 0 BAT: (printer(O)) (keyboard(I))
	1 1 UL1: (RS232C)	1 1 UP2: (none)	1 1 UR2: (none)	1 1 UC1: (RS232C(O)) (RS232C(I))

Device names in parentheses are those of the actual physical devices connected. The initial setting of IOBYTE is 10101001, that is LST: corresponds to the printer, PUN: to the RS232C port, RDR: to the RS232C port and CON: to both the CRT display (output) and keyboard (input).

File control block

Dr	F1	F2	/ /	F8	T1	T2	T3	Ex	S1	S2	Rc	D0	/ /	D15	Cr	R0	R1	R2
00	01	02	...	08	09	10	11	12	13	14	15	16	...	31	32	33	34	35

Dr: Drive code
- 0: Logged-in drive
- 1: Drive A
- 2: Drive B
- 5: Drive E
- 6: Drive F

F1 - F8: File name in uppercase ASCII
T1 - T3: File type in uppercase ASCII
Ex: Current extent number
S1: Reserved for system
S2: Reserved for system
Rc: Record count
D0 - D15: Assigned by CP/M.
Cr: Record counter
R0 - R2: Random access record counter

BIOS entry address

ADDRESS	ENTRY NAME
F600	BOOT
F603	WBOOT
F606	CONST
F609	CONIN
F60C	CONOUT
F60F	LIST
F612	PUNCH
F615	READER
F618	HOME
F61B	SELDSK
F61E	SETTRK
F621	SETSEC
F624	SETDMA
F627	READ
F62A	WRITE
F62D	LISTST
F630	SECTRN
F633	PSET
F636	HCOPY
F639	BEEP
F63C	RSOPEN
F63F	RSCLOSE
F642	RSINST
F645	RSOUTST
F648	RSIN
F64B	RSOUT
F64E	TIMDAT
F651	MEMORY
F654	RSIOX
F657	LIGHTPEN
F65A	MASKI
F65D	LOADX
F660	STORX
F663	LDIRX
F666	JUMPX
F669	CALLX
F66C	GETPFK
F66F	PUTPFK

QX-10 MFBASIC
Commands and statements

AUTO

FORMAT AUTO [<line number>[,[<increment>]]]

PURPOSE Initiates automatic generation of program line numbers.

EXAMPLE
AUTO
AUTO 100,50

BEEP

FORMAT BEEP <duration>
ON
OFF

PURPOSE Controls operation of the QX-10's sound generator.

EXAMPLE
BEEP ON
BEEP OFF
BEEP 100

BIT

FORMAT BIT ON | OFF

PURPOSE Specifies whether the character generator of the printer or QX-10 is to be used for printout.

EXAMPLE
BIT ON
BIT OFF

CALL

FORMAT CALL <variable name>[(<argument list>)]

PURPOSE Starts execution of a machine language subroutine.

EXAMPLE CALL I(X)

CHAIN

FORMAT CHAIN [MERGE]<filename>[,[<line number exp>]
[,ALL][,DELETE<range>]]

PURPOSE Chains execution of BASIC programs and passes variables from calling program to program called.

EXAMPLE CHAIN "SAMPLE"
CHAIN "A:SAMPLE3",,ALL
CHAIN MERGE "SUB",100

CIRCLE

FORMAT CIRCLE [STEP] (<horizontal position>, <vertical position>), <radius> [,[<color>][,[<starting angle>] [,[<ending angle>][,<ratio>]]]]

PURPOSE Draws circles, ellipses, or arcs.

EXAMPLE CIRCLE (100,100),50
CIRCLE STEP (10,15),50

CLEAR

FORMAT CLEAR[[<dummy1>][,[<,upper memory limit>] [,<dummy2>]]]

PURPOSE Clears all numeric and string variables. When options are specified, also reserves an area in memory for machine language programs.

EXAMPLE 10 CLEAR ,&HBFFF

CLOSE

FORMAT CLOSE[[#]<file number>[,[#]<file number...>]]

PURPOSE Terminates access to files.

EXAMPLE CLOSE #3

CLS

FORMAT CLS

PURPOSE Clears the display screen.

EXAMPLE CLS

COLOR

FORMAT COLOR [<foreground color>] [,<background color>]

PURPOSE Specifies the screen colors.

EXAMPLE COLOR 3,4

COMMON

FORMAT COMMON<list of variables>

FORMAT Passes variables to a CHAINed program.

PURPOSE COMMON B,A$

CONNECT

FORMAT CONNECT[STEP](X1,Y1)-[STEP](X2,Y2)[...[STEP](Xn,Yn)] [,[<color code>][,<line style>]]

PURPOSE Draws lines between specified points on the screen.

EXAMPLE CONNECT (0,0)-(20,20)
CONNECT STEP(0,0)-STEP(50,50)

CONT

FORMAT CONT

PURPOSE Resumes execution of a program interrupted by STOP, END, or the BREAK key.

EXAMPLE CONT

COPY

FORMAT COPY

PURPOSE Outputs the display contents to the printer.

EXAMPLE COPY

DATA

FORMAT DATA<list of constants>

PURPOSE Stores numeric and string constants which are substituted into variables by the READ statement.

EXAMPLE DATA QX,10,EPSON

DATE$

FORMAT DATE$="<MM>/<DD>/<YY>"

PURPOSE Sets the date of the QX-10's calendar clock.

EXAMPLE DATE$="01/28/83"

DAY

FORMAT DAY=<W>

PURPOSE Sets the day of the week of the QX-10's calendar clock.

EXAMPLE DAY=6

DEF FN

FORMAT DEF FN<name>[(<parameter list>)]=<function definition>

PURPOSE Defines and names user-written functions.

EXAMPLE DEF FNA(X,Y)=X*3/(Y+2)

DEFINT/SNG/DBL/STR

FORMAT DEF | INT / SGN / DBL / STR | <range(s) of letters>

PURPOSE Declares types of variables as integer, single precision, double precision, or string.

EXAMPLE DEFINT I-N,W-Z

DEF USR

FORMAT DEF USR[<digit>]=<integer expression>

PURPOSE Used to specify the starting addresses of user-written machine language subroutines.

EXAMPLE DEF USR1=&HC000

DELETE

FORMAT DELETE [<line number 1>][-<line number 2>]

FORMAT Deletes specified lines from the program.
DELETE 40
DELETE 40-100
DELETE -40

DIM

FORMAT DIM<list of subscripted variables>

PURPOSE Specifies the maximum range of array subscripts and allocates space for storage of array variables.

EXAMPLE DIM G%(25),F%(25)

EDIT

FORMAT EDIT <line number>

PURPOSE Enters the EDIT mode at the specified line.

EXAMPLE EDIT 40

END

FORMAT END

PURPOSE Terminates program execution, closes all files, and returns MFBASIC to the command level.

EXAMPLE END

ERASE

FORMAT ERASE<list of array names>

PURPOSE Cancels array definitions made with the DIM statement.

EXAMPLE ERASE A,B

ERROR

FORMAT ERROR<integer expression>

PURPOSE Simulates the occurrence of MFBASIC errors or allows the user to define his own error codes.

EXAMPLE ERROR 255

FIELD

FORMAT FIELD[#]<file number>,<field width> AS<string variable>,<field width> AS<string variable>,...

PURPOSE Assigns positions in a random file buffer for use as variables.

EXAMPLE FIELD #1,20 AS N$,10 AS ID$,40 AS ADD$

FILES

FORMAT FILES [<file descriptor>]

PURPOSE Displays the name of files stored on a flexible disk.

EXAMPLE FILES "L???????.BAS"
FILES "B:"
FILES "B:D???.*"

FOR...NEXT

FORMAT FOR<variable> = <expression 1> TO <expression 2> [STEP <expression 3>]

.
.

NEXT[<variable>][,<variable>...]

PURPOSE Repeats the series of instructions between FOR and NEXT a given number of times.

EXAMPLE FOR I = 1 TO 100 STEP 4

.
.
.

NEXT I

GCURSOR

FORMAT GCURSOR [STEP](horizontal position, vertical position),(<variable 1>, <variable 2>)

PURPOSE Displays the graphic cursor and reads its coordinates into variables.

EXAMPLE GCURSOR (X1,Y1),(X2,Y2)

GET

FORMAT GET[#]<file number>[,<record number>]

PURPOSE Reads a record from a random disk file.

EXAMPLE GET #1,X

GET@

FORMAT GET[@] (horizontal position 1,vertical position 1)-[STEP](horizontal position 2,vertical position 2),<array name>

PURPOSE Reads the settings of the specified range of display dots into a variable array.

EXAMPLE GET@ (50,50)-(300,150),A

GOSUB...RETURN

FORMAT GOSUB<line number>

.
.
.

RETURN

PURPOSE Used for branching to and returning from subroutines.

EXAMPLE GOSUB<line number>

.
.
.

RETURN

GOTO

FORMAT GOTO<line number>

PURPOSE Branches program execution to the program line specified by<line number>.

EXAMPLE GOTO 200

IF...THEN[...ELSE]/IF...GOTO[...ELSE]

FORMAT IF<logical expression>
THEN <statement> | <line No.>
GOTO <line No.>
[ELSE <statement> | <line No.>]

PURPOSE Changes the flow of program execution according to the result of a logical expression.

EXAMPLE IF A=B THEN PRINT "A=B" ELSE PRINT "A<>B"

INPUT

FORMAT INPUT[;]<list of variables>
INPUT[;]<"prompt string"><;|,><list of variables>

PURPOSE Allows values to be substituted into variables from the keyboard during program execution.

EXAMPLE INPUT "NAME";A$

INPUT

FORMAT INPUT #<file number>,<variable list>

PURPOSE Reads data into variables from a sequential file.

EXAMPLE INPUT #1,A$,B$,C$

KEYn

FORMAT KEY<n>,<X$>

PURPOSE Defines the functions of programmable function keys.

EXAMPLE KEY 2,"LIST"+CHR$(13)

KEY LIST/KEY LLIST

PURPOSE Outputs a list of the programmable function key definitions to the display or printer.

EXAMPLE KEY LIST

KILL

FORMAT KILL <file descriptor>

PURPOSE Deletes the specified disk file.

EXAMPLE KILL "FILE3.BAS"
KILL "B:SAMPLE1.BAS"

LET

FORMAT [LET] <variable> = <expression>

PURPOSE Assigns the value of an expression to a variable.

EXAMPLE PI = 3.14159
LET PI = 3.14159

LINE

FORMAT LINE [([STEP](X1,Y1)]-[STEP](X2,Y2) [,[<color code>][,[B[F]][,<line style>]]]

PURPOSE Draws a straight line between two specified points.

EXAMPLE LINE (0,0)-(500,300)
LINE -STEP(20,20)
LINE (25,25)-(500,200),4,,&HAAAA

LINE INPUT

FORMAT LINE INPUT[;][<"prompt string">;] <string variable>

PURPOSE Assigns character strings entered from the keyboard during program execution to variables.

EXAMPLE LINE INPUT "ENTER NAME (LAST,FIRST)";A$

LINE INPUT

FORMAT LINE INPUT # <file number>,<string variable>

PURPOSE Reads lines of data into variables from a sequential disk file.

EXAMPLE LINE INPUT #1,A$

LIST/LLIST

FORMAT LIST | LLIST [<line number>[-[<line number>]]]
LIST <file descriptor>[<line number>[-[<line number>]]]

PURPOSE Lists BASIC program lines on the display or printer, or to a specified file.

EXAMPLE LIST
LIST -50
LIST 50-
LIST 50-200
LIST "CMOS:"

LOAD

FORMAT LOAD <file descriptor>[,R]

PURPOSE Loads a program into memory.

EXAMPLE LOAD "LNINPT"
LOAD "B:LNINPT.BAS"

LOCATE

FORMAT LOCATE [<X>][,[<Y>][,<cursor switch>]]

PURPOSE Moves the cursor to specified coordinates on the screen.

EXAMPLE LOCATE 1,1,0

LPRINT

FORMAT LPRINT [<list of expressions>]

PURPOSE Outputs data to a printer connected to the QX-10.

EXAMPLE LPRINT "EPSON QX-10"

LPRINT USING

FORMAT LPRINT USING <"format string">; <list of expressions>

PURPOSE Outputs data to the printer in a specific format.

EXAMPLE LPRINT USING "####";A;B

LSET/RSET

FORMAT LSET <string variable> = <string expression>
RSET <string variable> = <string expression>

PURPOSE Prepares character data for storage in a random access file by moving it into a random file buffer.

EXAMPLE LSET A$ = B$

MERGE

FORMAT MERGE <file descriptor>

PURPOSE Merges a program from a file (disk, disk image RAM, COM0:-COM4:, or CMOS:) with the program currently in memory.

EXAMPLE MERGE "TEST1"
MERGE "CMOS:"

MID$

FORMAT MID$(<string exp1>,n[,m]) = <string exp2>

PURPOSE Replaces characters from position n in <string exp1> with the first m characters of <string exp2>.

EXAMPLE MID$(A$,5,7) = B$

NAME

FORMAT NAME <old filename> AS <new filename>

PURPOSE Changes the names of files on a flexible disk.

EXAMPLE NAME "SAMPLE1.BAS" AS "SAMPLE2.BAS"

NEW

FORMAT NEW

PURPOSE Deletes the program in memory and clears all variables.

EXAMPLE NEW

ON ERROR GOTO

FORMAT ON ERROR GOTO[<line number>]

PURPOSE Causes program execution to branch to the first line of an error handling subroutine when an error occurs.

EXAMPLE ON ERROR GOTO 1000

ON...GOSUB/ON...GOTO

FORMAT ON <expression> GOTO <list of line numbers>
ON <expression> GOSUB <list of line numbers>

PURPOSE Branches to one of several specified program line numbers, depending on the value returned for <expression>.

EXAMPLE ON A GOSUB 100,200,500,1000
ON ASC(A$) GOTO 300,500,900

OPEN

FORMAT OPEN "<mode>",[#]<file number>, <file descriptor>,[<reclen>]

PURPOSE Opens a disk file or other device for input or output.

EXAMPLE OPEN"O",#1,"CLIENTS.DAT"

OPTION BASE

FORMAT OPTION BASE $\begin{vmatrix}0\\1\end{vmatrix}$

PURPOSE Declares the minimum value for array subscripts.

EXAMPLE OPTION BASE 1

OPTION COUNTRY

FORMAT OPTION COUNTRY <character string>

PURPOSE Specifies the international character set to be used for keyboard input/output, CRT display, and output to the printer.

EXAMPLE OPTION COUNTRY "U"
OPTION COUNTRY "england"

OPTION CURRENCY

FORMAT OPTION CURRENCY<character expression>

PURPOSE Changes the currency symbol.

EXAMPLE OPTION CURRENCY "@"

OPTION STYLE

FORMAT OPTION STYLE<numeric expression>

PURPOSE Specifies the character font to be used for display of 1-byte characters in the WIDTH 40 screen mode.

EXAMPLE OPTION STYLE 16

OUT

FORMAT OUT<integer expression 1>,<integer expression 2>

PURPOSE Outputs the value of <integer expression 2> to the machine output port specified in <integer expression 1>.

PAINT

FORMAT PAINT [STEP](X,Y)[,[<area color>][,<border color>]]

PURPOSE Paints the area including (X,Y) and surrounded by <border color> with <area color>.

EXAMPLE PAINT (300,200),3,7
PAINT STEP (A,B),2,6

PEN

FORMAT PEN ON | OFF

PURPOSE Turns ON or OFF input from the light pen.

EXAMPLE PEN ON
PEN OFF

POKE

FORMAT POKE<integer expression 1>,<integer expression 2>

PURPOSE Writes the data byte specified by <integer expression 2> to the memory address specified by <integer expression 1>.

EXAMPLE POKE &HC000,A

PRESET

FORMAT PRESET [STEP](X,Y)[,<color code>]

PURPOSE Resets the dot at the specified graphic display coordinates.

EXAMPLE PRESET (X,Y)
PRESET STEP (10,10),7

PRINT

FORMAT PRINT [<expression>][;]
[,]
<expression>...][;]
[,]

PURPOSE Outputs data to the display screen.

EXAMPLE PRINT "Name is";A$
PRINT X,Y

PRINT USING

FORMAT PRINT USING<"format string">;<list of expressions>

PURPOSE Outputs data to the display screen in the format specified in <"format string">.

EXAMPLE PRINT USING "\ \";A$;B$;C$
PRINT USING "#####.##";A;B;C

PRINT

FORMAT PRINT #<file number>,<list of expressions>

PURPOSE Writes data to a sequential file.

EXAMPLE PRINT #1,A$;";";B$

PRINT # USING

FORMAT PRINT #<file number>,USING<"format string">;<list of expressions>

PURPOSE Writes data to a sequential file in a specific format.

EXAMPLE PRINT#1 USING "####.##";A;B;C

PSET

FORMAT PSET [STEP](X,Y)[,<color code>]

PURPOSE Sets the dot at the specified graphic coordinates.

EXAMPLE PSET (A,B)
PSET STEP (5,-5),4

PUT

FORMAT PUT[#]<file number>[,<record number>]

PURPOSE Writes a data record to a random access file.

EXAMPLE PUT#1,X

PUT@

FORMAT PUT[@](X,Y),<variable array name>[,<function>]

PURPOSE Displays a graphic pattern stored in memory on the display screen.

EXAMPLE PUT@(50,50),A#,PSET

RANDOMIZE

FORMAT RANDOMIZE [<expression>]

PURPOSE Reinitializes the random number generator.

EXAMPLE RANDOMIZE
RANDOMIZE VAL(RIGHT$(TIME$,2))

READ

FORMAT READ<list of variables>

PURPOSE Reads values from DATA statements and substitutes them into variables.

EXAMPLE READ A$,B$,C$

REM

FORMAT REM<remark>
' <remark>

PURPOSE Used to insert explanatory remarks into a program.

EXAMPLE ' REGRESSION ROUTINE

RENUM

FORMAT RENUM [[<new line number>][,[<old line number>][,<increment>]]]

PURPOSE Renumbers the lines of programs.

EXAMPLE RENUM
RENUM 300,50
RENUM 1000,900,20

RESET

FORMAT RESET

PURPOSE Resets the READ ONLY condition after the flexible disk in one of the drives has been replaced.

EXAMPLE RESET

RESTORE

FORMAT RESTORE [<line number>]

PURPOSE Resets the pointer which keeps track of the last item read from DATA statements.

EXAMPLE RESTORE
RESTORE 1000

RESUME

FORMAT RESUME
RESUME 0
RESUME NEXT
RESUME<line number>

PURPOSE Used to continue program execution after completion of an error processing routine.

EXAMPLE RESUME 100

RUN

FORMAT RUN [<line number>]
RUN<file descriptor>[,R]

PURPOSE Starts execution of a program.

EXAMPLE RUN 300
RUN "B:SAMPLE",R

SAVE

FORMAT SAVE<file descriptor>[,A | ,P]

PURPOSE Saves the program in memory to a disk file or CMOS RAM.

EXAMPLE
```
SAVE"ADDRESS.DAT"
SAVE"CMOS:"
```

SET

FORMAT SET<file descriptor>[,P]

PURPOSE Sets or resets the write protect attribute of disk files.

EXAMPLE
```
SET "B:CLIENTS.DAT",P
SET "B:CLIENTS.DAT"
```

SOUND

FORMAT SOUND<pitch>,<duration>

PURPOSE Outputs a tone of the specified pitch and duration from the speaker.

EXAMPLE SOUND 1000,100

STOP

FORMAT STOP

PURPOSE Terminates program execution and returns MFBASIC to the command level.

EXAMPLE STOP

STOP KEY

FORMAT STOP KEY | ON | OFF |

PURPOSE Disables or reenables the BREAK key.

EXAMPLE
```
STOP KEY ON
STOP KEY OFF
```

SWAP

FORMAT SWAP<variable 1>,<variable 2>

PURPOSE Exchanges the values of variables specified in <variable 1> and <variable 2>.

EXAMPLE SWAP A$,B$

SYSTEM

FORMAT SYSTEM

PURPOSE Clears program memory and returns control to the CP/M operating system.

EXAMPLE SYSTEM

TIME$

FORMAT TIME$="<HH>:<MM>:<SS>"

PURPOSE Sets the time of the QX-10's built-in clock.

EXAMPLE TIME$="15:35:00"

TRON/TROFF

FORMAT
```
TRON
TROFF
```

PURPOSE Starts or stops the trace mode of program execution.

EXAMPLE
```
TRON
TROFF
```

WAIT

FORMAT WAIT<port number>,I[,J]

PURPOSE Suspends program execution until a specified bit pattern is developed at the specified machine input port.

WHILE...WEND

FORMAT
```
WHILE<expression>
  .
  .
[<loop statements>]
  .
  .
WEND
```

PURPOSE Repeats the series of instructions included between WHILE and WEND as long as the result of the specified expression is TRUE.

EXAMPLE
```
WHILE A=<100 . .
WEND
```

WIDTH

FORMAT
```
WIDTH<no. of columns>
WIDTH<file descriptor>,<no. of columns>
WIDTH #<file no.>,<no. of columns>
WIDTH LPRINT<no. of columns>
```

PURPOSE Sets the column width of the specified device or file.

EXAMPLE
```
WIDTH 40
WIDTH "LPT0:",80
WIDTH #1,80
WIDTH LPRINT 40
```

WRITE

FORMAT WRITE[<list of expressions>]

PURPOSE Displays data on the CRT screen.

EXAMPLE WRITE A$,B$,C$

WRITE

FORMAT WRITE # <file number>,<list of expressions>

PURPOSE Writes data to a sequential disk file using the format of the WRITE statement.

EXAMPLE WRITE #1,A$,B$

Functions

ABS

FORMAT ABS(X)

PURPOSE Returns the absolute value of expression X.

EXAMPLE A = ABS(-10)

ASC

FORMAT ASC(X$)

PURPOSE Returns the numeric value which is the ASCII code for the first character of string X$.

EXAMPLE A = ASC("A")

ATTR$

FORMAT ATTR$ (<file descriptor>)

PURPOSE Returns the setting of the write protect attribute for the specified disk file.

EXAMPLE A$ = ATTR$("CLIENTS.DAT")

ATN

FORMAT ATN(X)

PURPOSE Returns the arc tangent in radians for X.

EXAMPLE A = ATN(0.5)

CDBL

FORMAT CDBL(X)

PURPOSE Converts numeric expression X to a double precision number.

EXAMPLE A# = CDBL(X!)

CHR$

FORMAT CHR$(I)

PURPOSE Returns the character whose ASCII code equals the value of integer expression I.

EXAMPLE A$ = CHR$(65)

CINT

FORMAT CINT(X)

PURPOSE Rounds the decimal portion of numeric expression X to the nearest whole number and returns the equivalent integer value.

EXAMPLE A% = CINT(5.6)

COS

FORMAT COS(X)

PURPOSE Returns the cosine of angle X, where X is in radians.

EXAMPLE A# = COS(1.570796326794897)

CSNG

FORMAT CSNG(X)

PURPOSE Converts numeric expression X to a single precision number.

EXAMPLE A! = CSNG(16%)

CSRLIN

FORMAT CSRLIN

PURPOSE Returns the vertical character coordinate of the cursor.

EXAMPLE A% = CSRLIN

CVI/CVS/CVD

FORMAT CVI(<2-byte string>)
CVS(<4-byte string>)
CVD(<8-byte string>)

PURPOSE Converts ASCII representations of BCD code to numeric values.

EXAMPLE
```
PRINT CVI(CHR$(5)+CHR$(0))
PRINT CVS(CHR$(0)+CHR$(0)+CHR$(32)+CHR$(131))
PRINT CVD(CHR$(0)+CHR$(0)+CHR$(0)+CHR$(0)+CHR$(0)+CHR$(32)+CHR$(131)
```

DATE/DATE$

FORMAT DATE
DATE$

PURPOSE Returns the date of the QX-10's built-in clock.

EXAMPLE
```
A% = DATE
A$ = DATE$
```

DAY

FORMAT DAY

PURPOSE Returns the day of the week from the QX-10's built-in clock.

EXAMPLE
```
A% = DAY
```

DSKF

FORMAT DSKF (<drive name>)

PURPOSE Returns the number of kilobytes of free area on the disk in the specified drive.

EXAMPLE
```
A% = DSKF("A:")
```

EOF

FORMAT EOF(<file number>)

PURPOSE Returns a value indicating whether the end of a sequential file has been reached during sequential input.

EXAMPLE
```
IF EOF(1) THEN 100
```

ERL

FORMAT ERL

PURPOSE Returns the line number of a command/statement causing an error during program execution.

EXAMPLE
```
B = ERL
```

ERR

FORMAT ERR

PURPOSE Returns the error code of errors occurring during command or statement execution.

EXAMPLE
```
A = ERR
```

EXP

FORMAT EXP(X)

PURPOSE Returns the value of the natural base e to the power of X.

EXAMPLE
```
A = EXP(X)
```

FIX

FORMAT FIX(X)

PURPOSE Returns the integer portion of numeric expression X.

EXAMPLE
```
A = FIX(X)
```

FONT

FORMAT FONT (<X$>)

PURPOSE Returns the font number of the first character in string expression X$.

EXAMPLE
```
A = FONT(X$)
```

FRE

FORMAT FRE(0)
FRE(X$)

PURPOSE Returns the number of unused bytes of text or string area memory.

EXAMPLE
```
PRINT FRE(0)
PRINT FRE(A$)
```

HEX$

FORMAT HEX$(X)

PURPOSE Returns a character string representing the hexadecimal value of X.

EXAMPLE PRINT HEX$(44323)

INKEY$

FORMAT INKEY$

PURPOSE Checks the keyboard buffer and returns one character (or a null string if no key has been pressed).

EXAMPLE A$ = INKEY$

INP

FORMAT INP(I)

PURPOSE Returns one byte of data from machine port I.

EXAMPLE A = INP(176)

INPUT$

FORMAT INPUT$(X[,[#]Y])

PURPOSE Reads a string of X characters from the keyboard buffer or file opened under file number Y.

EXAMPLE A$ = INPUT$(1)
A$ = INPUT$(10, #1)

INSTR

FORMAT INSTR([I,]X$,Y$)

PURPOSE Searches for string Y$ in string X$ and returns a number indicating the position at which it was found.

EXAMPLE A = INSTR(X$,"ABC")

INT

FORMAT INT(X)

PURPOSE Subtracts the decimal portion from X and returns the integer value which is the result.

EXAMPLE A = INT(-B/3)

LEFT$

FORMAT LEFT$(X$,I)

PURPOSE Returns a string of I characters from the left end of string X$.

EXAMPLE A$ = LEFT$(X$,5)

LEN

FORMAT LEN(X$)

PURPOSE Returns the number of characters in string X$.

EXAMPLE A = LEN(X$)

LOC

FORMAT LOC(<file number>)

PURPOSE Returns the random access file record number following that used by the last GET or PUT statement, or the number of file sectors read/written since a sequential file was opened.

EXAMPLE A = LOC(1)

LOF

FORMAT LOF(<file number>)

PURPOSE Returns the size of a file.

EXAMPLE A = LOF(1)

LOG

FORMAT LOG(X)

PURPOSE Returns the natural logarithm of X.

EXAMPLE PRINT LOG(2.7812818)

LPOS

FORMAT LPOS(X)

PURPOSE Returns the current position of the pointer in the printer output buffer. (X is a dummy argument.)

EXAMPLE A = LPOS(X)

MID$

FORMAT MID$(X$,I[,J])

PURPOSE Returns J characters from the middle of X$, starting with character I.

EXAMPLE A$ = MID$("ABCDEFG",3,3)

MKI$/MKS$/MKD$

FORMAT MKI$(<integer expression>)
MKS$(<single precision expression>)
MKD$(<double precision expression>)

PURPOSE Converts numeric values to strings for storage in random access files.

EXAMPLE A$ = MKI$(X%)
A$ = MKS$(X!)
A$ = MKD$(X#)

OCT$

FORMAT OCT$(X)

PURPOSE Returns a character string representing the octal value of X.

EXAMPLE A$ = OCT$(9999)

PEEK

FORMAT PEEK(I)

PURPOSE Returns one byte of data from memory location I.

EXAMPLE A = PEEK(&HE000)

PEN

FORMAT PEN(<function>)

PURPOSE Returns the status of the light pen.

PEN(0) - Returns -1 if the light pen has been triggered.

PEN(1) - Returns the horizontal graphic screen coordinate at which the light pen was triggered.

PEN(2) - Returns the vertical graphic screen coordinate at which the light pen was triggered.

PEN(3) - Returns the horizontal character coordinate at which the light pen was triggered.

PEN(4) - Returns the vertical character coordinate at which the light pen was triggered.

POINT

FORMAT POINT (horizontal coordinate,vertical coordinate)

PURPOSE Returns the color of the display dot at the specified graphic screen location.

EXAMPLE PRINT POINT(100,100)

POS

FORMAT POS(<file no.>)

PURPOSE Returns the current position of the pointer in the file output buffer.

EXAMPLE PRINT POS(1)

RIGHT$

FORMAT RIGHT$(X$,I)

PURPOSE Returns I characters from the right end of string X$.

EXAMPLE A$ = RIGHT$("abcdefg",3)

RND

FORMAT RND[(X)]

PURPOSE Returns a random number between 0 and 1.

EXAMPLE A = RND

SCREEN

FORMAT SCREEN (<horizontal position>,<vertical position> [,<function>])

PURPOSE Returns the code or character type of the character at the specified screen location.

EXAMPLE A = SCREEN(5,5)

SGN

FORMAT SGN(X)

PURPOSE Returns the sign of X.

EXAMPLE A = SNG(X)

SIN

FORMAT SIN(X)

PURPOSE Returns the sine of X.

EXAMPLE A = SIN(X)

SPACE$

FORMAT SPACE$(X)

PURPOSE Returns a string of spaces of a specified length.

EXAMPLE A$ = "AAA" + SPACE$(10) + "CCC"

SPC

FORMAT SPC(I)

PURPOSE Returns a string of I spaces for output to the display or printer.

EXAMPLE PRINT SPC(10);A$

SQR

FORMAT SQR(X)

PURPOSE Returns the square root of X.

EXAMPLE PRINT SQR(2#)

STR$

FORMAT STR$(X)

PURPOSE Returns a string representation of the value of X.

EXAMPLE A$ = STR$(123)

STRING$

FORMAT STRING$(I,J)
STRING$(I,X$)

PURPOSE Returns a string of I characters.

EXAMPLE PRINT STRING$(10,65)
PRINT STRING$(10,"A")

STYLE$

FORMAT STYLE$(<X$>,<font 1>,<font 2>)

PURPOSE Returns X$ as a character string in which characters in <font 1> have been converted to <font 2>.

EXAMPLE A$ = STYLE$(X$,2,4)

TAB

FORMAT TAB(I)

PURPOSE Moves the cursor (print head) to character position I on the display screen (print line).

EXAMPLE PRINT TAB(10);"ABC"
LPRINT TAB(10);"ABC"

TAN

FORMAT TAN(X)

PURPOSE Returns the tangent of X.

EXAMPLE A = TAN(3.1416/4)

TIME

FORMAT TIME

PURPOSE Returns the number of seconds which have passed since 00:00:00.

EXAMPLE A = TIME

TIME$

FORMAT TIME$

PURPOSE Returns the time of the QX-10's built-in clock.

EXAMPLE $ = TIME$

USR

FORMAT USR[<digit>](argument)

PURPOSE Calls a machine language subroutine defined by a DEF USR statement.

EXAMPLE A = USR0(B)

VAL

FORMAT VAL(X$)

PURPOSE Returns the numeric value of a string composed of numeric ASCII characters.

EXAMPLE A = VAL("123")

VARPTR

FORMAT VARPTR(<variable name>)
VARPTR(#<file number>)

PURPOSE Returns the address in memory of the specified variable or file buffer.

EXAMPLE PRINT HEX$(VARPTR(A))
PRINT HEX$(VARPTR(#1))