QP/M 2.7

Interface Guide

September, 1985

Created by
MICROCode Consulting
www.microcodeconsulting.com

CP/M 2.2 is a registered trademark of Digital Research, Incorporated. Z-80 is a registered trademark of Zilog, Incorporated. NSC-800 is a trademark of National Semiconductor Corporation. WordStar is a trademark of MicroPro International. Macro-80, Link-80, and Lib-80 are registered trademarks of MicroSoft Corporation.

DISCLAIMER

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

TABLE OF CONTENTS

INTRODUCTION

QP/M 2.7 is a superset of CP/M 2.2. It is intended to upgrade any CP/M system that uses a Z80 or NSC-800 microprocessor to a faster, more powerful operating system. Original CP/M packages include the CCP (Command Control Processor) and BDOS (Basic Disk Operating System). The CP/M 2.2 CCP has been totally replaced by an upgraded version called QCP that includes a most important TIME command; CP/M 2.2 BDOS has been replaced by QP/M 2.7 QDOS. Two unique features appear in QDOS: Drive/User Search whenever a file is opened, and automatic time/date stamping of files. CP/M 2.2 BIOS (Basic Input/Output System) need not be changed. In order to use the time/date capability, a new vector (called TIMDAT) must be added, either within BIOS or elsewhere, to return the current time and date. Specific procedures can be found in this documentation.

Function 0: System Reset

Entry Parameters	**Returned Value**
Register C: 00H	None

Function 0 *(System Reset)* returns control to the QP/M operating system at the QCP level. The QCP will re-initialize the disk subsytem by selecting and logging-in disk drive A. Function 0 has exactly the same effect as a jump to location BOOT (usually at memory location 0000H).

Function 1: Console Input

Entry Parameters	**Returned Value**
Register C: 01H	Register A: ASCII input character

Function 1 *(Console Input)* reads the next console character into register A. All printable characters, characters with high-bit set, carriage return, linefeed, and backspace (CTRL+H) will be echoed to the console output. Tab characters (CTRL+I), although returned as a single character, will be expanded in columns of eight characters by console output. A check is made for CTRL+S; when this occurs, program execution is suspended (screen display is frozen) until another character (except CTRL+C) is typed. Typing a CTRL+S immediately followed by a CTRL+C aborts program execution and forces a system reset. All other control characters are returned, but not echoed.

Function 2: Console Output

Entry Parameters	**Returned Value**
Register C: 02H Register E: ASCII output character	None

Function 2 *(Console Output)* sends the ASCII character in register E to the console. Tab characters (CTRL+I, ASCII value 9) are expanded in columns of eight characters. During Console Output, the console is checked for any characters waiting. The character CTRL+S is immediately processed as described in Console Input; all other characters are either discarded OR the first non-CTRL+S character is held for the next console input depending on the QDOS configuration (see Installation Guide).

Function 3: Reader Input

Entry Parameters	**Returned Value**
Register C: 03H	Register A: ASCII input character

Function 3 *(Reader Input)* reads the next character from the logical reader into register A. Control will not return until a character has been read.

Function 4: Punch Output

Entry Parameters	**Returned Value**
Register C: 04H Register E: ASCII output character	None

Function 4 *(Punch Output)* sends the character from register E to the logical punch device.

Function 5: List Output

Entry Parameters	**Returned Value**
Register C: 05H Register E: ASCII output character	None

Function 5 *(List Output)* sends the ASCII character in register E to the logical listing device.

Function 6: Direct Console I/O

Entry Parameters	**Returned Value**
Register C: 06H Register E: 0FFH (input) **-OR-** 0FEH (status) **-OR-** character (output)	Register A: status (input and status query only)

Function 6 *(Direct Console Input/Output)* is supported under QP/M for those specialized applications where unadorned console input and output is required. Use of this function, in general, should be avoided as it bypasses all of QP/M's normal character functions such as detection of CTRL+S and tab expansion.

Upon entry to function 6, register E either contains hexadecimal value 0FF, denoting console input request, hexadecimal value 0FE, denoting console status request, or any other character which denotes console output.

If the input value in register E is 0FFH, function 6 returns A = 00 if no character is ready, otherwise A contains the next console input character.

If the input value in register E is 0FEH, function 6 returns A = 00 if no character is ready or 0FFH if a character is ready; the character is not returned.

If the input value of E is neither 0FFH nor 0FEH, function 6 assumes that register E contains a valid ASCII character which is sent to the console.

Function 7: Get I/O Byte

Entry Parameters	**Returned Value**
Register C: 07H	Register A: I/O byte value

Function 7 *(Get I/O Byte)* returns the current value of the IOBYTE in register A.

Function 8: Set I/O Byte

Entry Parameters	**Returned Value**
Register C: 08H Register E: I/O byte value	None

Function 8 *(Set I/O Byte)* changes the system IOBYTE value to that given in register E.

Function 9: Print String

Entry Parameters	Returned Value
Register C: 09H Register DE: String address	Register DE: Address of terminating character

Function 9 *(Print String)* sends the character string stored in memory at the location given by register pair DE to the console device. Tabs are expanded as in function 2, and checks are made for CTRL+S. When the string terminator character ("$", ASCII hexadecimal value 24) is encountered, control is returned to the calling program. Upon exit, register pair DE points to the terminating character ("$" in QP/M).

Function 10: Read Console Buffer

Entry Parameters	Returned Value
Register C: 0AH Register DE: Buffer address	Console characters in buffer

Function 10 *(Read Console Buffer)* reads a line of edited console input into a buffer pointed to by register pair DE. Console input is terminated when either a Carriage Return (ASCII hexadecimal value 0D) or a Line Feed (ASCII hexadecimal value 0A) is input or the buffer is full. The input buffer pointed to by DE takes the following form:

Read Console Buffer

DE	+0	+1	+2	+3	+4	+5	+6	+7	+8	…	+n
	mx	Nc	c1	c2	c3	c4	c5	c6	c7	…	??

where "mx" is the maximum number of characters that the buffer will hold (a value between 1 and 255), "nc" is the number of characters actually read (set by QDOS before return), followed by the characters read from the Console Input. If "nc" < "mx", then uninitialized positions follow the last character, denoted by "??" in the above figure (as the value is not known).

The Read Console Buffer performs its function by reading characters from the Console Input, processing certain characters, and placing the rest into the buffer. If the buffer fills before the termination character (return or line feed) is encountered, QDOS returns to the calling program with the full buffer. Exception characters which are processed are not placed into the input buffer; these are listed below along with their function.

Character Function

ctrl-C	reboots (performs a System Reset) when encountered at the beginning of the line
ctrl-E	forces a physical end-of-line (Console Output is sent a new line character sequence) without terminating console input
ctrl-H	(or delete) backspaces one character erasing that character from the console and input buffer
ctrl-J	(line feed) terminates the input line
ctrl-M	(return) terminates the input line
ctrl-R	retypes the current line after new line character sequence is sent
ctrl-U	removes current line after new line
ctrl-X	backspaces to beginning of current line

Note also that certain functions which return the carriage to the leftmost position (e.g. ctrl-X) do so only to the column position where the Read Console Buffer function started.

QP/M does not support the "rubout echo" method for the delete/rubout key, but rather treats it identical to backspace. As all video terminals support either the backspace key or the rubout key and hardcopy-only terminals have faded, this feature should not be missed.

Function 11: Get Console Status

Entry Parameters	**Returned Value**
Register C: 0BH	Register A: Console status

Function 11 *(Get Console Status)* checks if a character has been typed at the console. If a character is ready, the hexadecimal value FF is returned in register A; otherwise, 00H is returned.

Function 12: Return Version Number

Entry Parameters	**Returned Value**
Register C: 0CH	Register A: 27H

Function 12 *(Return Version Number)* provides information which allows version-independent programming. A two-byte value is returned, with H = 00 designating CP/M and QP/M releases (H = 01 for MP/M), and L = 00 for all releases previous to 2.0. CP/M 2.0 returns a hexadecimal value 20 in register L, with subsequent version releases 21 and 22. Version number 27 designates QP/M. Using function 12, for example, you can write application programs which select between user-prompted time/date input or system fetch of time/date depending on which version of the operating system is resident.

Function 13: Reset Disk System

Entry Parameters	**Returned Value**
Register C: 0DH	None

Function 13 *(Reset Disk System)* is used to restore the file system to a reset state where all disks are set for both read and write, only disk drive A is selected, and the default DMA address is reset to BOOT+0080H. This function can be used by an application program that requires a disk change without a system reboot.

Function 14: Select Disk

Entry Parameters	**Returned Value**
Register C: 0EH Register E: Disk number (0=A, 1=B, ..., 15=P)	None

Function 14 *(Select Disk)* function designates the disk drive named in register E as the default disk for subsequent file operations, with E = 00H for drive A, 01H for drive B, and so-forth through 0FH (15 decimal) for drive P. The drive is placed in an "on-line" status which subsequently activates that disk's directory until the next cold start, warm start, or disk system reset operation. If the disk media is changed while it is on-line, the drive immediately goes to a read/only status; a cold start, warm start, Reset Disk System, or Reset Drive function is required to return the disk to a read/write status.

The disk number specified is register E is different from the drive code specified in the FCB. FCB's which specify drive code zero (dr = 00H) automatically reference the currently selected drive. Drive code values of 1 through 16, however, will cause QP/M to ignore the selected default drive, and directly reference drives A through P. (Note that values 0 through 15 reference drives A through P in the Disk Select function.)

Function 15: Open File

Entry Parameters	Returned Value
Register C: 0FH Register DE: FCB address	Register A: Directory code

Function 15 *(Open File)* is used to activate a file which currently exists in the disk directory. QDOS scans the referenced disk directory for a match in positions 1 through 14 of the FCB referenced (bytes "s1" and "s2" are automatically zeroed), where an ASCII question mark ("?" - value 3FH) matches any directory character in any of these positions. For normal Open File operations, no question marks are included and, further, byte "ex" of the FCB is set to zero before calling this function.

Byte 0 of the FCB specifies the drive code. If "dr" = 00H, the default disk drive is selected. Drive codes "dr" = 01H through 10H auto-select drives A through P, respectively.

The default drive/user search feature within QDOS is ONLY activated when the Open File function is called.

During an Open File operation, QDOS first searches the disk directory of the current user area. If the file is not found, QDOS then searches the directory of the default user area for a match (unless the two user areas are identical OR default user search is disabled). However, a file not found in the current user area, but found in the default user area, can only be opened for reading. Any attempt to write to a file that was opened under the default user area will produce the message

```
 QDOS error on d: File R/O
```

The FCB of a file which is opened under the default user area is tagged by QDOS for future disk operations. As a result, a file can be read even if the current user area and/or the default user area changes after a successful open. This tag also prevents disk write operations to the file even if the current user area has been changed to match the user number of the file. To write to the file, it must be re-opened in the user area where that file exists.

If QDOS fails to find the file in either the current or default user area, it searches for the file on the default drive in the default user area (unless the default drive and current drive are the same OR default drive search is disabled). All restrictions described for default user search also apply to default drive search.

If a directory element is matched, the relevant directory information is copied into bytes *d0* through *dn* of the FCB, thus allowing access to the files through subsequent read and write operations. (Write operations are not allowed on files opened in the default drive and/or user area.) It should be noted that an existing file must not be accessed until a successful open operation is completed. Upon return, the Open File function returns a **directory code** with the value 0 through 3 if the open was successful, or 0FFH (decimal 255) if the file cannot be found. If question marks are present in the FCB, then the first matching FCB is activated. Also note that

(Function 15 continued)

the current record ("cr") must be zeroed by the program if you want to access the file sequentially from the first record.

The **FCB Format** is shown below.

DE	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12
	dr	f1	f2	f3	f4	f5	f6	f7	f8	t1	t2	t3	ex

DE	+13	+14	+15	+16	+17	+18	+19	+20	+21	+22	+23	+24	+25
	s1	s2	rc	d00	d01	d02	d03	d04	d05	d06	d07	d08	d09

DE	+26	+27	+28	+29	+30	+31	+32	+33	+34	+35
	d10	d11	D12	d13	d14	d15	cr	r0	r1	r2

Bytes	**Name**	**Function**
00	dr	Drive code (current drive=0, 1=A, 2=B, etc.)
01-08	f1-f8	File name
09-11	t1-t3	File extension
12	ex	Current extent number
13	s1	Used in QP/M for default drive/user tag (not used in CP/M)
14	s2	Set to 0 on call to Open, Make, and Search; used in QP/M for read/write tag
15	rc	Current extent record count
16-31	d00-d15	Disk map
32	cr	Current record for read/write
33-35	r0-r2	Random record number

Function 16: Close File

Entry Parameters	Returned Value
Register C: 10H Register DE: FCB address	Register A: Directory code

Function 16 *(Close File)* performs the inverse of the Open file function. If the FCB addressed by DE has been previously activated through an Open or Make function, the close function permanently records the new FCB in the referenced disk directory. The FCB matching process for the close is the same as the Open function. The directory code returned for a successful close operation will be 0, 1, 2, or 3, while a 0FFH (decimal 255) will be returned if the file name cannot be found in the directory.

A file need not be closed if only read operations have taken place. If write operations have occurred, however, the close operation is necessary to permanently record the new directory information.

Closing a file which was opened under the default drive and/or user area will return a directory code of 0FFH (decimal 255), except when the file drive and user areas are the same as the current drive and user area. In either case, no close operation is needed and the result is of no concern.

Function 17: Search for First

Entry Parameters	**Returned Value**
Register C: 11H Register DE: FCB address	Register A: Directory code

Function 17 *(Search for First)* scans the directory for a match with the file given by the FCB addressed by DE. The 0FFH (decimal 255) is returned if the file is not found, otherwise 0, 1, 2, or 3 is returned indicating the file is present. In the case where the file is found, the current DMA buffer is filled with the record containing the directory entry, and relative starting position is A (Directory code) * 32 (i.e., ADD A five times or RRC three times). Although not normally required for application programs, you can extract the directory information from the buffer at this position, if desired.

An ASCII question mark ("?", decimal 63, hexadecimal 3F) in any position "f1" though "ex" will match the corresponding field of any directory entry on the either the default or auto-selected disk drive. If the "dr" field contains an ASCII question mark, then the auto-disk select function is disabled and the default disk is searched returning any entry, allocated or free, belonging to any user number. (Normally, Search for First searches only the current user area.) If the "dr" field is not a question mark, the "s2" byte is automatically zeroed and search is done on the default or auto-selected disk in the current drive/user area.

Function 18: Search for Next

Entry Parameters	**Returned Value**
Register C: 12H	Register A: Directory code

Function 18 *(Search for Next)* is similar to the Search for First function, except that the directory scan continues from the last matched entry. Similiar to function 17, function 18 returns the decimal value 255 when no more directory items match.

Function 19: Delete File

Entry Parameters	**Returned Value**
Register C: 13H Register DE: FCB address	Register A: Result code

Function 19 *(Delete File)* removes the files which match the FCB addressed by register pair DE. The filename and type may contain ambiguous references (i.e. question marks in various positions), but the drive select code cannot be ambiguous, as in Search for First and Search for Next functions.

Function 19 returns a 0FFH (decimal 255) if the referenced file or files cannot be found in the current user area, otherwise a value of 00H is returned.

Function 20: Read Sequential

Entry Parameters	**Returned Value**
Register C: 14H Register DE: FCB address	Register A: Directory code

Function 20 *(Read Sequential)* reads the next 128-byte record from a file into memory at the current DMA address, provided a previous Open or Make function activated the FCB addressed by register pair DE. The record is read from position "cr" of the extent, and the "cr" field is incremented to the next record position automatically. If the "cr" field overflows, then the next logical extent is opened automatically and the "cr" field is reset to zero in preparation for the next read operation. The value 00H is returned in the A register if the read operation was successful, while a non-zero value is returned if the read operation failed (read past end-of-file, etc.).

If the file referenced has been opened in the default drive and/or user area, the read operation will automatically access the appropriate drive and/or user.

Function 21: Write Sequential

Entry Parameters	Returned Value
Register C: 15H Register DE: FCB address	Register A: Directory code

Function 21 *(Write Sequential)* writes the next 128-byte record at the current DMA address, provided a previous Open or Make function activated the FCB addressed by register pair DE. The record is read from position "cr" of the extent, and the "cr" field is incremented to the next record position automatically. If the "cr" field overflows, then the next logical extent is opened automatically and the "cr" field is reset to zero in preparation for the next write operation. The value 00H is returned in the A register if the write operation was successful, while a non-zero value is returned if the write operation failed (e.g. full disk or directory).

Write operations can take place into an existing file, in which case newly written records overlay those which already exist in the file.

Note that a file opened under the default drive and/or user area can only be read. Any attempt to write will result in the message

```
QDOS error on d: File R/O
```

In order to write to this file, one must re-open the file under the drive/user area in which that file resides.

Function 22: Make File

Entry Parameters	**Returned Value**
Register C: 16H Register DE: FCB address	Register A: Directory code

Function 22 *(Make File)* is similar to the Open File operation except that the FCB must name a file which does not exist in the currently referenced disk directory and user area (i.e., the one named explicitly by a non-zero "dr" code, otherwise the default disk). QDOS creates the file and initializes both the directory and main memory value to an empty file.

The programmer must ensure that no duplicate file names occur, and a preceding Delete File operation is sufficient if there is any possibility that a duplication could occur. Upon return, register A = 0, 1, 2, or 3 if the operation was successful and 0FFH (decimal 255) if unsuccessful due to no more directory space.

The Make function also activates the FCB, thus a subsequent Open File is not necessary.

Function 23: Rename File

Entry Parameters	**Returned Value**
Register C: 22H Register DE: FCB address	Register A: Error code

Function 23 *(Rename File)* uses the FCB addressed by DE to change all occurrences of the file named in the first 16 bytes to the file named in the second 16 bytes. The drive code "dr" at position 0 is used to select the drive, while the drive code for the new file name at position 16 is ignored - the drive is the same as the first file name (at position 0). Upon return, register A = 00H if the rename was successful, and 0FFH (decimal 255) if the first file name could not be found.

Again, caution should be used in renaming files just as in making files as no check is made to see if the file already exists. Having two files of the same name not only confuses QP/M, but renders the "individual" files useless.

Function 24: Return Login Vector

Entry Parameters	**Returned Value**
Register C: 18H	Register HL: Login vector

Function 24 *(Return Login Vector)* returns a 16-bit value in register pair HL, where the least significant bit of L corresponds to drive A and the most significant bit of H corresponds to drive P. A "0" bit indicates that the drive is not on-line, while a "1" bit marks a drive that is actively on-line due to an explicit disk drive selection or an implicit select specified by a non-zero "dr" field of an FCB.

Function 25: Return Current Disk

Entry Parameters	**Returned Value**
Register C: 19H	Register A: Current disk

Function 25 *(Return Current Disk)* returns the currently selected default disk number in register A. The disk numbers range from 0 for drive A to 15 for drive P.

Function 26: Set DMA Address

Entry Parameters	**Returned Value**
Register C: 1AH Register DE: DMA address	None

Function 26 *(Set DMA Address)* sets the DMA address to the value specified in register pair DE. "DMA" is an acronym for Direct Memory Access, which although is not what actually happens in QP/M, is accurate for programming purposes. The DMA address in QP/M designates the location where the 128-byte data record resides before a disk write and after a disk read. The DMA address is automatically set to BOOT+0080H upon cold start, warm start, or disk system reset. The Set DMA function, however, can be used to change this value to address another area of memory. This new DMA address is used for all subsequent operations until the next Set DMA function, cold start, warm start, or disk system reset.

Function 27: Get Allocation Address

Entry Parameters	**Returned Value**
Register C: 1BH	Register HL: Allocation address

Function 27 *(Get Alloc)* returns the address of the allocation vector. The "allocation vector" is maintained in main memory for each on-line disk drive. Function 27 will return the base address of the allocation vector for the currently selected disk drive; however, the allocation information may be invalid if the selected disk drive has been marked read/only.

Function 28: Write Protect Disk

Entry Parameters	**Returned Value**
Register C: 1CH	None

Function 28 *(Write Protect Disk)* provides temporary write protection for the currently selected disk. Any attempt to write to this disk prior to a reboot, Function 13 (Reset Disk), or Function 37 (Reset Drive) produces the message:

```
QDOS error on d: R/O
```

Function 29: Get Read/Only Vector

Entry Parameters	**Returned Value**
Register C: 1DH	Register HL: R/O vector value

Function 29 *(Get Read/Only Vector)* returns a bit vector in register pair HL that indicates which drives have the temporary read/only bit set. Similar to Return Login Vector (function 24), the least significant bit of register L corresponds to drive A and the most significant bit of register H corresponds to drive P. The R/O bit is set either by an explicit call to function 28, or by mechanisms within QP/M that detect disk changes.

Function 30: Set File Attributes

Entry Parameters	Returned Value
Register C: 1EH Register DE: FCB address	Register A: Directory code

Function 30 *(Set File Attributes)* allows programmatic manipulation of permanent indicators attached to files. Specifically, the R/O, System, and Archive attributes (t1', t2', and t3') can be set or reset. Register pair DE addresses an unambiguous file name (no question marks) with the appropriate attributes set or reset. Function 30 searches for a match, then changes the matched directory entry to contain the selected indicators. Indicators f1' through f4' are not presently used, but may be useful for applications programs, since they are not involved in the matching process during file operations. Indicators f5' through f8' are reserved for future systems.

Function 31: Get DPB Address

Entry Parameters	**Returned Value**
Register C: 1FH	Register HL: DPB address

Function 31 *(Get DPB Address)* returns the address of the BIOS resident disk parameter block in register HL; this address has two explicit applications: disk parameter values can be extracted for display and space computation purposes, or transient programs can dynamically change the values of current disk parameters when the disk environment changes, if required. Normally, application programs will not use this function.

Function 32: Set/Get User Code

Entry Parameters	**Returned Value**
Register C: 20H Register E: 0FFH (get) **-OR-** User code (set)	Register A: User code (get only)

Function 32 *(Set/Get User Code)* allows an application program to change or interrogate the currently active user number by calling function 32. If register E = 0FFH, then the value of the current user number is returned in the A register, which ranges from 0 to 31. If register E is not 0FFH, the current user number is then changed to the value in E (modulo 32).

Function 33: Read Random

Entry Parameters	**Returned Value**
Register C: 21H Register DE: FCB address	Register A: Error code

Function 33 *(Read Random)* is similar to the sequential file read operation except that the read operation takes place at a particular record number selected by the 24-bit value constructed from the three byte field following the FCB (byte positions r0 at 33, r1 at 34, and r2 at 35). This 24-bit value is stored with least significant byte first (r0), middle byte next (r1), and most significant byte last (r2). QP/M does not reference byte "r2", except in computing the size of the file. Byte "r2" must always be zero, since a non-zero value indicates overflow past the end of the file.

Thus, the r0/r1 byte pair is treated as a "word" value, which contains the record to read. This value ranges from 0 to 65535 (0000H to FFFFH) providing access to any particular record of the maximal 8 megabyte file. Before you can process a file using random access, the base extent (extent 0) must first be opened. Although the base extent may or may not contain any allocated data, this ensures that the file is properly recorded in the directory and is visible in directory requests. The selected record number is then stored into the random record field (r0/r1), and QDOS is called to read the record.

Upon return from a Read Random call, register A contains an error code (listed below) with 00H indicating the operation was successful. In the latter case, the current DMA address contains the randomly accessed record. Note that contrary to the sequential read operation, the record number (rc) is not advanced. Thus, subsequent random read operations continue to read the same record unless the bytes r0/r1 are changed.

Upon each Read Random operation, the logical extent and current record values are automatically set. Therefore, the file can be sequentially read or written, starting from the current randomly accessed position. Note, however, that in this case, the last randomly read record will be re-read as you switch from random mode to sequential read, and the last record will be re-written as you switch to a sequential write operation. Of course, you can simply advance the random record position following each random read or write to obtain the effect of a sequential I/O operation.

(continued)

The error codes returned in register A following a random read are as follows:

00	successful completion
01	reading unwritten data
02	(not used in random mode)
03	cannot close current extent
04	seek to unwritten record
05	(not returned in read mode)
06	seek past physical end of disk

Error codes 01 and 04 occur whenever a random read operation accesses a data block which has not been previously written, or an extent which has not been created; both are equivalent conditions. Error code 03 does not normally occur under proper system operation, but can be cleared by simply re-reading, or re-opening extent zero, provided you are not using a physically write-protected disk. Error code 06 occurs whenever byte "r2" is non-zero under the current QDOS release. Normally, non-zero return codes can be treated as missing data, with a zero return code indicating success.

Function 34: Write Random

Entry Parameters	**Returned Value**
Register C: 22H Register DE: FCB address	Register A: Error Code

Function 34 *(Write Random)* is initiated similar to the Read Random call, except that data is written to the disk from the current DMA address. Further, if the disk extent or data block that is the target of the write has not yet been allocated, the allocation will performed automatically before the write operation continues. Like the Read Random operation, the random record number is not changed as a result of the write. The logical extent number and current record positions of the FCB are set to correspond to the random record that is being written.

Again, sequential read or write operations can commence following a random write, provided the currently addressed record is either read or rewritten again as the sequential operation begins. Also like Function 33, you can simply advance the random record position following each write to get the effect of a sequential write operation.

The error codes returned by a random write are identical to the Read Random operation, except that error code 02 is not used, and error code 05 is used to indicate that a new extent cannot be created due to directory overflow.

00	successful completion
01	reading unwritten data
02	(not used in random mode)
03	cannot close current extent
04	seek to unwritten record
05	cannot create new extent
06	seek past physical end of disk

Function 35: Compute File Size

Entry Parameters	**Returned Value**
Register C: 23H Register DE: FCB address	Random record field set

Function 35 *(Compute File Size)* is called to obtain the size of the file specified.

When computing the size of a file, the register pair DE contains the address of the FCB in random mode format (bytes "r0", "r1", and "r2" are present). The FCB contains an unambiguous file name to use for the directory scan. Upon return, the random record bytes contain the record address of the record following the end of the file (virtual file size). If function 35 is called, and the byte "r2" is 01, then the file contains the maximum possible record count of 65536. Otherwise, bytes r0/r1 constitute a 16-bit value (r0 is the least significant byte and r1 is the most significant byte) which is the file size in 128-byte records.

Data can be appended to the end of an existing file by calling function 35 to set the random record position to the end of file, then performing a sequence of random writes starting at the preset record position.

The virtual size of a file corresponds to the physical size only when the file was written sequentially. If the file was created in random mode and "holes" exist so that it is not continuous, then the file may contain fewer records than the size indicates. For example, if only the last record of a 8 megabyte file was written in random mode (record number 65535), the virtual size is 65536 records, although only one block of data has actually been written.

Function 36: Set Random Record

Entry Parameters	Returned Value
Register C: 24H Register DE: FCB address	Random record field set

Function 36 *(Set Random Record)* causes QDOS to automatically produce the random record position from a file that has been read or written sequentially to a particular point. QDOS examines the current extent and record number and produces the logical 16-bit record number in "r1" and "r0". "r0" contains the low part of the word, "r1" contains the high part of the word, and "r2" is set to zero.

A use for function 36 is when you switch from a sequential read or write over to random read or write. A file is sequentially accessed to a particular point in the file, a call to function 36 sets the record number, and subsequent random file operations continue from this point in the file.

Function 37: Reset Drive

Entry Parameters	**Returned Value**
Register C: 25H Register DE: Reset Vector	None (Selected drives are reset)

Function 37 *(Reset Drive)* is used to selectively reset drive or drives currently on-line. This is an alternative to the Reset Disk System function allowing one to change disks without having to reset the DMA address and re-select the specific drive.

The register pair DE contains a 16-bit vector which designates which drives to reset; similar to earlier functions, the least significant bit of E designates drive A and the most significant bit of D designates drive P. A "1" in the specific bit position resets that drive; a "0" has no effect on the drive status.

Function 37 can be treated as a selective alternative to function 13 (Reset Disk System) or as the complement of function 28 (Write Protect Disk).

Function 38: Get Time/Date

Entry Parameters	**Returned Value**
Register C: 26H Register DE: FCB address **-OR-** 0000H	Register A: Directory code

Function 38 *(Get Time/Date)* returns either the current time/date or the time/date information for an unambiguous file name pointed to by the FCB. If register pair DE contains 0000H, QDOS returns the current time/date. If register DE is non-zero, it is assumed to point to a valid file name and retrieves the time/date information of that file.

When interrogating the current time/date (register DE = 0000H), QDOS fills the first 6 bytes of the DMA buffer with the current time/date information and returns a directory code of zero. The format of the current time/date is

DMA values when **current** time/date requested

DMA	**+0**	**+1**	**+2**	**+3**	**+4**	**+5**
	day	month	Year	hour	minute	second

(Day values range from 1 to 31, months from 1 to 12, years from 0 to 99, hours from 0 to 23, minutes and seconds from 0 to 59.)

If the operating system has not been configured for QP/M (i.e. time/date capability added), QDOS will fill the 6 bytes with zeroes.

(continued)

(Function 38 continued)

When interrogating the time/date of a valid file name (register DE = FCB address), QDOS fills the DMA buffer with the record containing time/date information and returns a directory code in register A = 0, 1, 2, 3, 4, 5, 6, 7 decimal; QDOS returns 0FFH (decimal 255) if either the file is not found or the disk is not time/date stamped. This directory code specifies the relative starting position of the time/date information and is A * 16 (i.e., ADD A four times) bytes from the start of the DMA buffer.

The format of the time/date information for a file is slightly different than the current time/date vector as is shown on the next page.

Offset from start of file time/date entry
(at Dir Code times 16 plus DMA address)

DMA	**+0**	**+1**	**+2**	**+3**	**+4**	**+5**	**+6**	**+7**	**+8**	**+9**	**+10**	**+11**	**+12**	**+13**	**+14**	**+15**
	Created					*Updated*					*Archived*					
	day	mon	yr	hr	Min	day	mon	yr	hr	min	day	mon	yr	hr	min	00H

Byte positions 0 through 5 specify the time/date the file was created under QP/M. Byte positions 6 through 9 contain the time/date of the last successful close following a file write operation while operating under QP/M. (Note that a QP/M disk operating under CP/M will not have this information updated.) Although not normally required for applications programs, bytes 10 through 14 specify the time/date of the last QBACKUP successfully completed on this file.

When QDOS returns a value of 0FFH (255), then either that file does not exist on disk or the QP/M time/date file information is absent. Note that a disk must have been configured for operation under QP/M before time/date stamping takes place (via QSTAMP.COM or QSTAMPX.COM or QSTAMPV.COM).

Function 39: Set File Time/Date

Entry Parameters	**Returned Value**
Register C: 27H Register DE: FCB address	Register A: Directory code

Function 39 *(Set File Time/Date)* performs the opposite function of function 38 (get time/date) on the designated file. Before calling this function, the DMA area should be initialized to the proper time/date information in the following format

DMA	**+0**	**+1**	**+2**	**+3**	**+4**	**+5**	**+6**	**+7**	**+8**	**+9**	**+10**	**+11**	**+12**	**+13**	**+14**	**+15**
	Created					*Updated*					*Archived*					
	day	mon	yr	hr	Min	day	mon	yr	Hr	min	day	mon	yr	hr	min	00 H

Upon completion, QDOS returns the directory code in register A. Successful completion values range from A = 0 to 7 decimal while 255 decimal (0FFH) designates an unsuccessful operation. Unsuccessful completion occurs when either the file does not exist or the specified disk has not been time/date stamped for QP/M.

Note that the original time/date information for the specific file is destroyed and completely replaced by the information in the DMA area. Care should be taken when using this function.

Function 40: Write Random Zero Fill

Entry Parameters	Returned Value
Register C: 28H Register DE: FCB Address	Register A: Error Code

Function 40 *(Write Random Zero Fill)* is similar to function 24 (Write Random) except that no data is actually written to the disk, but the block is allocated for future use. Of course, one could also use the Write Random (or Write Sequential) functions and obtain the same results. However, use of function 40 to allocate disk space is quicker as only a "write directory" function is performed instead of both "write data" and "write directory".

The error codes returned by a Random Write are identical to the Write Random operation and are included below:

00	successful completion
01	reading unwritten data
02	(not used in random mode)
03	cannot close current extent
04	seek to unwritten record
05	cannot create new extent
06	seek past physical end of disk

Function 41: Set/Get Default Drive

Entry Parameters	**Returned Value**
Register C: 29H Register E: 0FFH (get) -OR- Drive Code (set)	Register A: Drive Code (get)

Function 41 *(Set/Get Default Drive)* can be called by an application program to change or interrogate the current default drive number. If register E = 0FFH, then the value of the current default drive number is returned in the A register, which ranges from 0 to 14 (A=0, B=1, etc). If default drive search is disabled, a value of 0FFH is returned.

If the value in register E is not 255, then the default drive number is changed to the specified value in the E-register. Note that all values of default drive between 0 and 254 are accepted; however, only values between 0 and 14 activate the default drive search. **ALL other default drive codes DISABLE the default drive search feature of QDOS.**

The default drive number is only used by QDOS during an Open File operation.

Function 42: Set/Get Default User

Entry Parameters	**Returned Value**
Register C: 2AH Register E: 0FFH (get) **-OR-** User code (set)	Register A: User code (get only)

Function 42 *(Set/Get Default User)* can be called by an application program to change or interrogate the currently active default user number. If register E = 0FFH, then the value of the current default user number is returned in the A register, which ranges from 0 to 14 if active or 0FFH if default user search is disabled. If register E is not 0FFH, then the current default user number is changed to the specified value in the E-register. Note that all values of default user between 0 and 254 are accepted; however, only values between 0 and 14 activate the default user search. **ALL other default user codes DISABLE the default user search feature of QDOS.**

The default user number is only used by QDOS during an Open File operation.

Function 43: Compute Free Space

Entry Parameters	**Returned Value**
Register C: 2BH	Register HL: Number of free records (128 bytes)

Function 43 *(Compute Free Space)* returns the free space remaining on the current logged drive. The value returned is in physical records (128-bytes). This vector could be used, for example, by utility applications to determine if the remaining space on the logged drive is adequate for transferring or storing a file. If you wish to determine the remaining space on another drive, you must first use the Select Disk (function 14) followed by a call to Compute Free Space.