

### FIXMAP Command

In order to keep track of which blocks of disk space are occupied and which are free, TurboDOS maintains on each disk an allocation map for that disk's space. Certain program malfunctions (e.g., failure to close a newly-created file) can create discrepancies between the allocation map and the directory of a disk. The result is that disk blocks may occasionally become unavailable.

The FIXMAP command enables you to regenerate the allocation map for a disk, thereby reclaiming any disk blocks that may have become unavailable in this fashion. The command format is:

#### FIXMAP file

If the "file" argument is present, it specifies the desired disk drive. Otherwise, the logged-on drive is referenced.

In a multi-user system, you cannot use FIXMAP on a drive that is in use by another user.

#### Example:

```
0A) FIXMAP B:  
Drive B disk map re-initialized  
0A) FIXMAP C:  
Drive C disk map re-initialized, 3 allocation block(s) gained  
0A)
```

# User's Guide to TurboDOS

Copyright (C) 1981 by Software 2000 Inc.

## Commands

### FORMAT Command

The **FORMAT** command enables you to initialize or verify a disk. The command format is:

**FORMAT** drive ;options

where "drive" specifies the disk to be initialized. The command normally initializes and verifies the disk, but the ";V" option may be used to cause verification only. If the ";R" option is present, then **FORMAT** repeats (allowing multiple removable disks to be initialized or verified) until terminated by typing **CTRL-C**.

Other options on the **FORMAT** command are hardware-dependent. For floppy disks, the following options are available:

1 or 2	one- or two-sided diskette
S or D	single- or double-density
T or C	TurboDOS or CP/M-compatible format
3, 4 or 8	For minifloppies: 35-, 40- or 80-track disks

In a network configuration, **FORMAT** may not be used in a slave processor to access a drive owned by the master processor. However, a slave console may be attached to the master processor (see **MASTER** command) for the purpose of running **FORMAT**.

#### Examples:

```
0A) FORMAT B: ;2DT
```

```
Insert disk to be formatted in drive B
```

```
Enter <cr> to begin formatting: <cr>
```

```
Starting format pass
```

```
.....
```

```
Starting verify pass
```

```
.....
```

```
0A)
```

**NOTE:** On systems with several types of disk storage units, several hardware-dependent versions of the **FORMAT** command (called **FMTxxx**) will generally be provided.

### LABEL Command

The LABEL command enables you to record a volume label on any disk. The command format is:

LABEL label

The "label" argument has the same format as a file specification. The name and type fields are used as a volume label for the specified drive. If no drive is specified explicitly, then the logged-on drive is labelled. The label recorded on a disk may be displayed using the DIR or DRIVE command.

#### Examples

```
0A) LABEL B:PAYABLES.DAT
```

```
Disk label written.
```

```
0A) DRIVE B:
```

```
Disk characteristics, drive B:PAYABLES.DAT
```

```
(...etc...)
```

```
0A)
```

# User's Guide to TurboDOS

Copyright (C) 1981 by Software 2000 Inc.

## Commands

### LOGON Command

The LOGON command provides password-type security in multi-user configurations of TurboDOS. The purpose of this command is to prevent unauthorized access to the system and to protect private file libraries. The command format is:

#### LOGON

in response to which you are prompted to enter your user-ID from the console keyboard. The user-ID is validated against the file USERID.SYS in the user 31 library. USERID.SYS is an ASCII text file containing entries of the form:

userid, [password], userno[P], [drive]

where "userid" and "password" are up to 8 characters in length, "userno" is a user number 0...30, and "drive" is a drive letter A...P. The password and drive fields are optional. If your user-ID has an associated password specified in USERID.SYS, then LOGON prompts you to enter a password, and validates it. The log-on succeeds only if you enter both the user-ID and password correctly, in which case your console is logged onto the specified user number, and the specified drive is selected as the default disk. If your entry in USERID.SYS has the user number suffix "P", you are logged-on as a "privileged" user, enabling you to access various protected facilities of TurboDOS.

If the user 31 library also contains a file named SYSLOG.SYS, then the LOGON command will automatically record your log-on in that file.

### LOGOFF Command

The LOGOFF command is used in multi-user configurations of TurboDOS to terminate your session. The command format is:

#### LOGOFF

The LOGOFF command sets the user number to a reserved value (31) and selects the system drive as the default disk. The library for user 31 normally contains only the LOGON.COM command file and the USERID.SYS validation file. Consequently, no further activity can be performed until a successful LOGON has been accomplished.

If the library for user 31 on the system drive also contains a file named "SYSLOG.SYS", then the LOGOFF command will automatically record your log-off in that file.

#### Examples:

```
5C)LOGOFF  
31A}
```

**MASTER Command**

If you are using a slave console in a networking TurboDOS configuration, the MASTER command enables you to attach your console to the master processor. The command format is:

**MASTER**

which attaches your console to the master processor. To detach from the master processor (and resume normal slave console operation), enter an Attention-Abort sequence (CTRL-S CTRL-C).

While attached to the master, you can make attention requests of the master processor by using CTRL-A (instead of the usual CTRL-S).

NOTE: The MASTER command requires that the master operating system be generated with a special console driver module (CONREM).

Example:

```
3B)MASTER
Console attached to master processor
Enter User-ID: SYSTEM
Enter Password: SECRET
0A)BACKUP A: B:
.
.
.
0A)LOGOFF
Enter User-ID: <CTRL-S CTRL-C>
Console detached from master processor
3B)
```

## PRINT Command

The PRINT command enables you to control the routing of your print output. The command format is:

PRINT keyword keyword...

where "keyword" is chosen from the following list:

LOCAL  
REMOTE  
PRINTER=p  
QUEUE=q  
DRIVE=d  
FILE  
CONSOLE  
OFF

and "p", "q" and "d" are printer, queue and drive letters in the range A...P. Any keyword may be abbreviated to a single letter ("P" for PRINTER, "Q" for QUEUE, etc.).

PRINT DRIVE=d QUEUE=q causes print output to be spooled to print files on the specified drive, then queued automatically on the specified print queue for de-spooled printing.

PRINT DRIVE=d FILE causes print output to be spooled to print files on the specified drive, but not queued automatically for printing.

PRINT PRINTER=p causes print output to be routed directly to the specified printer without intermediate spooling to disk.

PRINT CONSOLE causes print output to be routed to the console.

PRINT OFF causes print output to be discarded.

PRINT without keywords causes the current print routing to be displayed.

In a networking system, you can route your printed output to either local (slave-owned) or remote (master-owned) printers by using the keywords LOCAL or REMOTE in any PRINT command.

# User's Guide to TurboDOS

Copyright (C) 1981 by Software 2000 Inc.

## Commands

### Examples

0A) PRINT D=C O=A

Printing is to LOCAL SPOOLER on DRIVE C to QUEUE A

0A}

0A) PRINT P=B

Printing is to LOCAL PRINTER B

0A}

0A) PRINT C

Printing is to LOCAL CONSOLE

0A}

0A) PRINT O

Printing is to LOCAL OFFLINE

0A}



### PRINTER Command

The PRINTER command enables you to control de-spooled printing. The command format is:

PRINTER p keyword keyword...

where "p" is a printer letter in the range A...P which specifies the printer to be controlled, and "keyword" is chosen from the following list:

LOCAL  
REMOTE  
QUEUE=q  
OFFLINE  
STOP  
GO  
BEGIN  
TERMINATE

Any keyword may be abbreviated to a single letter ("T" for TERMINATE, "Q" for QUEUE, etc.).

PRINTER p QUEUE=q causes the specified printer to take its print jobs from the specified de-spool queue. If the printer is currently printing from another queue, then the new assignment takes effect at the end of the current print job.

PRINTER p OFF causes the specified printer to be taken offline at the end of the current print job. An offline printer may be accessed for direct printing.

PRINTER p STOP temporarily suspends de-spooling to the specified printer (e.g., to correct a paper jam).

PRINTER p GO resumes de-spooling to the specified printer.

PRINTER p BEGIN stops de-spooling to the specified printer, and causes the current print job to be reprinted from the beginning when de-spooling is resumed.

PRINTER p TERMINATE terminates the current print job on the specified printer, and continues with the next queued job. The terminated print file is not deleted from disk, however, so the job may be manually requeued with the QUEUE command.

# User's Guide to TurboDOS

Copyright (C) 1981 by Software 2000 Inc.

## Commands

PRINTER p with no keywords displays the current status of the specified printer.

In a networking system, a PRINTER command run in a slave processor defaults to REMOTE unless the LOCAL keyword is included in the command.

### Examples:

0A) PRINTER B Q=A

LOCAL Printer B Assigned to QUEUE A

0A)

0A) PRINTER B S

LOCAL Printer B Assigned to QUEUE A (Stopped)

0A)

## QUEUE Command

The QUEUE command enables you to manually queue print files for de-spooled printing. The command format is:

QUEUE file ;options

where "file" specifies the file to be queued. Wild-card characters ("?" and "\*\*") may be used in the "file" argument to indicate that multiple files are to be queued.

If "file" is omitted from the command line, then the QUEUE command operates in an interactive mode. It reads successive directives from the console (prompted by an asterisk "\*\*"). A null directive terminates the command. The format of each interactive directive is:

file ;options

The options argument may include:

"Y" or "N" to specify whether or not you want to confirm each individual file before it is queued. If neither is specified and "file" contains wild cards, then the QUEUE command prompts you to specify whether or not you want confirmation.

"S" or "D" to specify whether you want the queued files to be saved or deleted after printing. If neither is specified, then the files are saved.

"Q=q" (where q is a queue letter in the range A...P) to specify which de-spool queue to use. If not specified, the current queue (as specified in the last PRINT command) is used. If there is no current queue in effect, then the "Q=q" option is required.

"L" or "R" to specify whether you want printing to be local or remote (meaningful only from a network slave). If neither is specified, current print routing (as specified in the last PRINT command) is used.

# User's Guide to TurboDOS

Copyright (C) 1981 by Software 2000 Inc.

## Commands

### Examples

```
0A) QUEUE -PRINT- * :NDO=A
```

```
A:-PRINT-008 queued
```

```
A:-PRINT-014 queued
```

```
A:-PRINT-020 queued
```

```
0A)
```

### RECEIVE Command

The RECEIVE command enables you to read one record from a FIFO, and to display it on the console. The command format is:

RECEIVE file

where "file" specifies the FIFO you wish to read. This command may be useful where FIFOs are used as message mailboxes in a multi-user system.

Example:

```
0A]RECEIVE MAILBOX.RMN
DICK...NEED TO MEET WITH YOU...JEFF
0A]
```

Note that RECEIVE displays only one record. If you want to display or print the entire contents of a FIFO, use the TYPE command instead.

## User's Guide to TurboDOS

Copyright (C) 1981 by Software 2000 Inc.

### Commands

#### RENAME Command

The RENAME command enables you to rename individual disk files or groups of files. The command format is:

```
RENAME oldfile newfile ;option
```

where "oldfile" and "newfile" specify the old and new file names. Wild-card characters ("?" and "\*\*") may be used in the "oldfile" argument to indicate that multiple files are to be renamed. Wild-card characters may also be used in the "newfile" argument to indicate that the corresponding characters of each old file name are to be used in naming each new file. An "oldfile" or "newfile" argument consisting only of a drive letter (e.g., "C:") implies a file specification of all wild cards (i.e., "C:\*.\*\*").

If both "oldfile" and "newfile" are omitted from the command line, then the RENAME command operates in an interactive mode. It reads successive directives from the console (prompted by an asterisk "\*\*"). A null directive terminates the command. The format of each interactive directive is:

```
oldfile newfile ;option
```

The "option" argument may be either "Y" or "N" (preceded by a semicolon), and specifies whether or not you want to confirm each individual file before it is renamed. If "option" is omitted and "oldfile" contains wild cards, then the RENAME command prompts you to specify whether or not confirmation is desired.

Examples:

0A) RENAME ;N

\* \*.BAK \*.PRV

A:REFERENC.BAK renamed to A:REFERENC.PRIV

A:USERGUID.BAK renamed to A:USERGUID.PRIV

\* \*.TXT \*.BAK

A:LETTER .TXT renamed to A:LETTER .BAK

A:REFERENC.TXT renamed to A:REFERENC.BAK

A:USERGUID.TXT renamed to A:USERGUID.BAK

\*

0A}

0A) RENAME B:MAXI\*.\* OPT\*.\* ;Y

OK to rename B:MAXICOMP.TXT to B:OPTICOMP.TXT (y/n)? N

OK to rename B:MAXIMUMS.COM to B:OPTIMUMS.COM (y/n)? Y

B:MAXIMUMS.COM renamed to B:OPTIMUMS.COM

OK to rename B:MAXIMUMS.FOR to B:OPTIMUMS.FOR (y/n)? Y

B:MAXIMUMS.FOR renamed to B:OPTIMUMS.FOR

OK to rename B:MAXIMUMS.REL to B:OPTIMUMS.REL (y/n)? N

0A}

# User's Guide to TurboDOS

Copyright (C) 1981 by Software 2000 Inc.

## Commands

### RESET Command

The RESET command enables a privileged user to reset one or more network slave processors from any terminal of a networking TurboDOS system. Up to 16 slaves are denoted by the letters A...P, and the command format is:

RESET slaves

where "slaves" is a string of one or more slave letters A...P, "^" if only the requesting slave is to be reset, or "\*" if all slaves are to be reset.

Examples:

```
0A)RESET D
```

```
0A)
```

```
0A)RESET ^
```

```
Enter User-ID:
```



**SEND Command**

The SEND command enables you to write one record to a FIFO. The command format is:

**SEND file message**

where "file" specifies the FIFO you wish to write, and "message" is any text you wish. This command may be useful where FIFOs are used as batch queues or message mailboxes in a multi-user system.

Examples:

```
0A)SEND BATCH.DO DO COMPLINK BIGPROG
0A)SEND MAILBOX.RMN DICK...NEED TO MEET WITH YOU...JEFF
0A)
```

# User's Guide to TurboDOS

Copyright (C) 1981 by Software 2000 Inc.

## Commands

### SET Command

The SET command enables you to set and clear the various file attributes, and has the following format:

SET file ;options +onattributes -offattributes

where "onattributes" and "offattributes" consist of any set of the mnemonic letters E, S, F, R, G, and A (corresponding to the attributes Exclusive, Shared, FIFO, Read-only, Global and Archived). Attributes following the "+" are set, attributes following the "-" are cleared, and all other attributes are left unchanged.

If "file" contains wild-cards, the SET command can modify the attributes of multiple files; the usual "Y" or "N" options may be used to specify whether or not you want to confirm individual files. If "file" is omitted from the command line, then the SET command operates in interactive mode.

Example:

```
0A)SET *.COM ;N +RG -S
```

sets all .COM files on the default drive to Read-only, Global, and not Shared.

If "file" contains only a drive specification, then the SET command may be used to set a disk volume read-only or read/write.

Example:

```
0A)SET B: ;+R
```

Drive B set to read-only

```
0A)SET B: ;-R
```

Drive B set to read/write

```
0A)
```

### SHOW Command

The SHOW command enables you to display the settings of file attributes. The command format is:

SHOW file ;options

If "file" contains wild-cards, the SHOW command can display the attributes of multiple files. If "file" is omitted from the command line, then the SET command operates in interactive mode. "Options" may include ;Y or ;N to specify whether or not to pause after displaying each file, or ;L to print rather than display.

For example:

0A)SHOW B:\*.\* :L

prints a listing of file attributes for all files on the B-drive.

If "file" contains only a drive specification, then the SHOW command may be used to determine whether a disk volume is read-only or read/write.

Example:

```
0A)SHOW A:
Drive A set to read-only
0A)SHOW B:
Drive B set to read/write
0A)
```

**User's Guide to TurboDOS**  
Copyright (C) 1981 by Software 2000 Inc.  
Commands

**TYPE Command**

The TYPE command enables you to display the contents of an ASCII file on the console or printer. The command format is:

TYPE file ;L

where "file" specifies the disk file to be displayed. If the ";L" suffix is present, then the contents of the file is printed. Otherwise, it is displayed on the console.

**Examples:**

```
0A|TYPE B:TURBODOS.DOC  
...ASCII display...  
0A|
```

USER Command

The USER command enables you to change the current user number. This command is honored only if you have logged-on as a privileged user. The command format is:

USER usernumber

where "usernumber" specifies the desired user number (between 0 and 31). If the "usernumber" argument is omitted, then the current user number is simply displayed (it also appears in the command prompt).

Examples:

```
0A} USER 2  
Current user number: 2  
2A} USER 0  
Current user number: 0  
0A}
```

**VERIFY Command**

The VERIFY command enables you to scan a volume (typically a hard-disk) for bad blocks and to mark them so that TurboDOS will avoid their use. The command format is:

**VERIFY drive**

where "drive" specifies the disk to be verified. If bad blocks are detected, the command will create a read-only directory entry with the filename BLOCKS.BAD to mark the defective blocks.

NOTE: the bad-block marking function of the VERIFY command presently works only in a single-user configuration of TurboDOS.

## CP/M-COMPATIBLE FUNCTIONS

All TurboDOS functions are invoked by means of a call to 0005H with a function number in the C-register. Invalid function numbers result in no operation.

### Function 0: Return to Operating System

**Called with:** C = 0

**Returns with:** (does not return)

**Notes:** This function (also called "warm start") is used to terminate execution of a transient program. The same function is more commonly performed by executing a jump to location 0000H, which has exactly the same effect.

### Function 1: Input Character from Console

**Called with:** C = 1

**Returns with:** A = input character

**Notes:** The console input character is echoed to console output.

### Function 2: Output Character to Console

**Called with:** C = 2  
E = output character

### Function 3: Input Character from Reader

**Called with:** C = 3

**Returns with:** A = input character

**Notes:** If the reader device is not implemented, this function performs no operation.

# **User's Guide to TurboDOS**

Copyright (C) 1981 by Software 2000 Inc.

CP/M-Compatible Functions

## **Function 4: Output Character to Punch**

**Called with:** C = 4  
E = output character

**Notes:** If the reader device is not implemented, this function performs no operation.

## **Function 5: Output Character to Printer**

**Called with:** C = 5  
E = output character

**Notes:** If the spooler is activated, the character is actually output to a print file.

## **Function 6: Direct Console Input/Output**

**Called with:** C = 6  
E = -1 (for console input)  
output character (for console output)

**Returns with:** A = input character  
0 if no input available

**Notes:** If the E-register contains a value other than -1, then that value is sent to the console device as an output character. If the E-register contains -1, then any available console input character is returned in the A-register. If no console input character is available, then 0 is returned in the A-register. This function does not echo console input characters to console output.

## **Function 7: Return I/O Assignments**

**Called with:** C = 7

**Returns with:** A = I/O Assignment Byte

**Notes:** This function simply returns the value of RAM location 0003H.



Function 8: Set I/O Assignments

Called with: C =  
E = I/O Assignment Byte

Notes: This function simply sets the value of RAM location 0003H. It has no effect whatever on TurboDOS I/O assignment.

Function 9: Output Buffer to Console

Called with: C = 9  
DE = buffer address

Notes: A string of characters terminated by "\$" is output to the console.

Function 10: Input Buffer from Console

Called with: C = 10  
DE = buffer address

Notes: The first byte of the buffer must be preset to the maximum number of characters to be input. Console input is accepted until terminated by a carriage-return. Input errors may be corrected by using backspace or delete characters to erase one character, and CTRL-U or CTRL-X to erase the entire line. On return, the second byte of the buffer contains the actual number of characters input from the console. The input string is returned starting in the third byte of the buffer.

Function 11: Return Console Status

Called with: C = 11

Returns with: A = -1 if console input is available  
0 if console input not available

**Function 12: Return O/S Version**

**Called with:** C = 12

**Returns with:** HL = 0022H (CP/M Version 2.2)  
BA = 0022H (CP/M Version 2.2)  
D = environment vector:  
    bit 7 = 1 for master processor  
    bit 6 = 1 for privileged log-on  
    bits 3-0 = slave processor ID (0...15)  
E = 1xH (TurboDOS Version 1.x)

**Notes:** The TurboDOS version number is returned in the E-register, and the environment vector in the D-register. The latest compatible CP/M version number is returned in the HL- and BA-registers.

**Function 13: Reset Disk System**

**Called with:** C = 13

**Returns with:** A = 0

**Notes:** In TurboDOS, the only effect of this function is to reset the record buffer address to 0080H.

**Function 14: Select Default Drive**

**Called with:** C = 14  
E = default drive (0="A", 1="B", ..., 15="P")

**Function 15: Open File**

**Called with:** C = 15  
DE = FCB address

**Returns with:** A = 0 if successful  
                  -1 if file not found

**Function 16: Close File**

**Called with:** C = 16  
DE = FCB address

**Returns with:** A = 0 if successful  
-1 if file not found

**Function 17: Search for First File**

**Called with:** C = 17  
DE = FCB address

**Returns with:** A = 0, 1, 2 or 3 if successful  
-1 if file not found

**Notes:** Returns with directory record (four entries) in record buffer. A-register specifies which of the four directory entries was found.

**Function 18: Search for Next File**

**Called with:** C = 18

**Returns with:** A = 0, 1, 2 or 3 if successful  
-1 if file not found

**Notes:** Continues search initiated by function 17. Continues from last directory entry found. Returns with directory record (four entries) in record buffer. A-register specifies which of the four directory entries was found.

**Function 19: Delete File**

**Called with:** C = 19  
DE = FCB address

**Returns with:** A = 0 if successful  
-1 if file not found

**Notes:** If FCB contains wild-cards, all matching files are deleted.

# User's Guide to TurboDOS

Copyright (C) 1981 by Software 2000 Inc.

CP/M-Compatible Functions

## Function 20: Read Next Record

**Called with:** C = 20  
DE = FCB address

**Returns with:** A = 0 if successful  
1 if at end-of-file  
2 if reading unwritten data  
8 if reading locked record

## Function 21: Write Next Record

**Called with:** C = 21  
DE = FCB address

**Returns with:** A = 0 if successful  
1 if file too large (>134 MB)  
2 if disk full or write-protected  
8 if writing locked record  
-1 if no directory space

## Function 22: Create File

**Called with:** C = 22  
DE = FCB address

**Returns with:** A = 0 if successful  
-1 if no directory space

## Function 23: Rename File

**Called with:** C = 23  
DE = FCB address

**Returns with:** A = 0 if successful  
-1 if no file found

**Notes:** The first 16 bytes of the FCB contains the file reference to be renamed, and the next 16 bytes contains the new file reference to be used. Wild-cards in the first reference may cause multiple files to be renamed. Wild-cards in the second reference cause the corresponding character positions to remain

unchanged.

**Function 24: Return Ready Vector**

**Called with:** C = 24

**Returns with:** HL = ready vector

**Notes:** This function replaces the "Return Login Vector" function of CP/M. (TurboDOS does not require disks to be logged-in before use.) The ready vector contains a one-bit for each drive that is ready, and a zero-bit for each drive that is not ready. The least significant bit corresponds to drive "A", and the most significant bit to drive "P". This function makes it possible for programs to scan all ready drives in search of a file.

**Function 25: Return Default Drive**

**Called with:** C = 25

**Returns with:** A = default drive (0="A", 1="B", ..., 15="P")

**Function 26: Set Record Buffer Address**

**Called with:** C = 26

DE = record buffer address

**Notes:** The record buffer address (sometimes called the "DMA address") is set to 0080H by default, unless this function is used to change it to another value.

**Function 27: Return Disk Allocation Information**

**Called with:** C = 27

E = drive (0="A", 1="B", ..., 15="P")

**Returns with:** A = block size (3=1K, 4=2K, ..., 7=16K)

C = directory blocks

DE = remaining blocks

HL = total blocks

**Notes:** This function is not compatible with CP/M. (Function 27 in CP/M is used only by the STAT utility.)

**Function 28: Write Protect Drive**

**Called with:** C = 28

**Notes:** Write protects the default drive until the next warm-start.

**Function 29: Return Write Protect Vector**

**Called with:** C = 29

**Returns with:** HL = write-protect vector

**Notes:** The write-protect vector contains a one-bit for each drive that is write-protected, and a zero-bit for each drive that is not. The least significant bit corresponds to drive "A", and the most significant bit to drive "p".

**Function 30: Set File Attributes**

**Called with:** C = 30  
DE = FCB address

**Returns with:** A = 0 if successful  
-1 if file not found

**Notes:** This function searches the directory for the file referenced by the FCB, and updates the file attributes in the directory from those in the FCB. The file attributes are stored in the sign (parity) bits of FCB bytes 2 through 12.

**Function 31: Return Physical Disk Information**

**Called with:** C = 31  
E = drive (0="A", 1="B", ..., 15="p")

**Returns with:** A = sector size (0=128, 1=256, 2=512, ..., 7=16K)  
BC = reserved tracks  
DE = tracks per disk  
HL = sectors per track

**Notes:** This function is not compatible with CP/M. (Function 31 in CP/M is used only by the STAT utility.)

**Function 32: Set/Return User Number**

**Called with:** C = 32  
E = user number 0...31 (or -1)

**Returns with:** A = user number 0...31

**Notes:** If the E-register contains the value -1, then the current user number is returned in the A-register. If the E-register contains some other value and the user is privileged, then the the current user number is changed to that value (modulo 32).

**Function 33: Read Random Record**

**Called with:** C = 33  
DE = FCB address

**Returns with:** A = 0 if successful  
1 if reading unwritten data  
3 if error changing extents  
4 if reading unwritten extent  
6 if random record number too large  
8 if reading locked record

**Notes:** This function reads the record number specified by the random record number field (FCB bytes 34 through 36).

**Function 34: Write Random Record**

**Called with:** C = 34  
DE = FCB address

**Returns with:** A = 0 if successful  
2 if disk full or write-protected  
3 if error changing extents  
5 if no directory space  
6 if random record number too large  
8 if writing locked record

**Notes:** This function writes the record number specified by the random record number field (FCB bytes 34 through 36).

**Function 35: Compute File Size**

**Called with:** C = 35  
DE = FCB address

**Returns with:** A = 0 if successful  
-1 if file not found

**Notes:** This function computes the size of the file referenced by the FCB, and returns the number of records in the random record number field (FCB bytes 34 through 36).

**Function 36: Set Random Record**

**Called with:** C = 36  
DE = FCB address

**Notes:** This function returns the current file position in the random record number field (FCB bytes 34 through 36). Since the sequential access functions (20 and 21) do not update the random record number field, function 36 should be used when switching from sequential to random access.



**Function 37: Reset Write-Protect Vector**

**Called with:** C = 37  
DE = reset vector

**Notes:** This function write-enables the drives which correspond to one-bits in the reset vector. The least significant bit corresponds to drive "A", and the most significant bit to drive "p".

**Function 40: Write Random Record with Zero Fill**

**Notes:** TurboDOS treats function 40 as a synonym for function 34.

# User's Guide to TurboDOS

Copyright (C) 1981 by Software 2000 Inc.

CP/M-Compatible Functions

## File Control Block Organization

The File Control Block (FCB) consists of an area of 33 bytes for sequential file access, or 36 bytes for random file access. The FCB organization is shown below:

<u>Bytes</u>	<u>Field</u>	<u>Description</u>
1	Drive	0=default, 1="A", 2="B", ..., 16="P"
2...9	Name	file name in ASCII, right-padded with spaces sign bits reserved for file attributes
10...12	Type	file type in ASCII, right-padded with spaces sign bits reserved for file attributes
13	Extent	least significant 5 bits of extent number
14	Flags	used by TurboDOS (Do Not Use)
15	Extent'	most significant 8 bits of extent number
16	Record Count	number of records in current extent 0...128 (Do Not Use)
17-32	Map	allocation map of current extent (Do Not Use)
33	Current Record	current record number 0...127 in current extent used and incremented by sequential read and write
34-36	Random Record	20-bit record number used by random read and write byte 34 is least significant

Simulated CP/M BIOS Branch Table

TurboDOS provides a full simulated "BIOS branch table" in order to support programs which make direct calls on the CP/M BIOS. This branch table always begins 255 bytes below the top of memory in order to ensure that it begins on a page boundary (as it does in CP/M). The entry points are:

xF00H	JMP	BOOT	; Cold Start
xF03H	JMP	WBOOT	; Warm Start
xF06H	JMP	CONST	; Console Status to A-reg
xF09H	JMP	CONIN	; Console Input to A-reg
xF0CH	JMP	CONOUT	; Console Output from C-reg
xF0FH	JMP	LIST	; Printer Output from C-reg
xF12H	JMP	PUNCH	; Punch Output from C-reg
xF15H	JMP	READER	; Reader Input to A-reg
xF18H	JMP	HOME	; Set Track to Zero
xF1BH	JMP	SELDSK	; Select Disk Drive from C-reg
xF1EH	JMP	SETTRK	; Set Track from BC-regs
xF21H	JMP	SETSEC	; Set Sector from BC-regs
xF24H	JMP	SETDMA	; Set DMA Address from BC-regs
xF27H	JMP	READ	; Read Disk Sector
xF2AH	JMP	WRITE	; Write Disk Sector
xF2DH	JMP	LISTST	; List Status to A-reg
xF30H	JMP	SECTRN	; Sector Translate Subroutine

The disk read and write entrypoints are honored only for privileged users.



## ADDITIONAL FUNCTIONS

Additional TurboDOS functions use function numbers in the range 64...127, and are intended primarily for use by TurboDOS commands and internal processes. These functions may be used safely by user programs, except for those marked "Do Not Use".

### Function 83: Set Date & Time

**Called with:**

- C = 83
- HL = Julian date
- D = hours (0...23)
- E = minutes (0...59)
- B = seconds (0...59)

**Notes:** The Julian date in HL should be the number of days since the base date of December 31, 1947. Dates prior to the base date are represented by negative numbers.

### Function 84: Return Date & Time

**Called with:** C = 84

**Returns with:**

- HL = Julian date
- D = hours (0...23)
- E = minutes (0...59)
- B = seconds (0...59)

**Notes:** The Julian date in HL is be the number of days since the base date of December 31, 1947. Dates prior to the base date are represented by negative numbers.

**Function 85: (Reserved)**

**Notes:** Do Not Use.

**Function 86: Physical Disk Access**

**Called with:** C = 86  
DE = PDR packet address

**Returns with:** A = 0 or -1

**Notes:** This function is available for privileged log-ons only. It provides direct access to the physical disk drivers, in order to support the BACKUP and FORMAT commands. On entry, the DE-registers contain the address of a 14-byte Physical Disk Request (PDR) packet. The first byte of the PDR packet is an operation code which determines the physical operation to be performed. Refer to the disk driver interface specification for details.

**Function 87: Return Comm Channel Status**

**Called with:** C = 87  
D = channel number

**Returns with:** A = 0 if comm channel input not available  
A = -1 if comm channel input is available

**Function 88: Input Character from Comm Channel**

**Called with:** C = 88  
D = channel number

**Returns with:** A = input character

Function 89: Output Character to Comm Channel

Called with: C = 89  
D = channel number  
E = output character

Function 90: Set Comm Channel Baud Rate

Called with: C = 90  
D = channel number  
E = baud rate codes  
bit 7 = 1 if attention detection enabled  
bit 6 = 1 if clear-to-send handshaking enabled  
bit 5 = 1 if output-only (input disabled)  
bits 3-0 = baud-rate value 0...15 (see table below)

**Notes:** The least significant nibble of the E-register contains a baud rate value as follows:

0 =	50	8 =	1,800
1 =	75	9 =	2,000
2 =	110	10 =	2,400
3 =	134.5	11 =	3,600
4 =	150	12 =	4,800
5 =	300	13 =	7,200
6 =	600	14 =	9,600
7 =	1,200	15 =	19,200

Function 91: Return Comm Channel Baud Rate

Called with: C = 91  
D = channel number

Returns with: A = baud rate code

**Notes:** A-register returns a baud rate code as described for function 90.

# User's Guide to TurboDOS

Copyright (C) 1981 by Software 2000 Inc.

## Additional Functions

### Function 92: Set Modem Controls

**Called with:**

- C = 92
- D = channel number
- E = modem control vector:
  - bit 7 = request-to-send
  - bit 6 = data-terminal-ready

### Function 93: Return Modem Status

**Called with:**

- C = 93
- D = channel number

**Returns with:**

- A = modem status vector:
  - bit 7 = clear-to-send
  - bit 6 = data-set-ready
  - bit 5 = data-carrier-detect
  - bit 4 = ring-indicator

### Function 94: (Reserved)

**Notes:** Do Not Use.

### Function 95: (Reserved)

**Notes:** Do Not Use.

### Function 96: Set Buffer Parameters

**Called with:**

- C = 96
- D = number of buffers
- E = buffer size (0=128, 1=256, 2=512, 7=16K)

**Notes:** Number of buffers must be at least 2. Buffer size must be at least as large as the largest physical disk sector size in use. This function causes all buffers to be written out (if necessary) and placed on the free chain.



Function 97: Return Buffer Parameters

Called with: C = 97

Returns with: A = memory size (pages, 0=64K)  
H = number of buffers  
L = buffer size (0=128, 1=256, 2=512, 7=16K)

Function 98: Activate DO-File

Called with: C = 98  
DE = FCB address (or 0)

Returns with: A = 0 if successful  
-1 if file not found

**Notes:** Only the first 16 bytes of the FCB are used. The file need not have been previously opened. This function causes any currently-active DO-file and command line to be stacked, and a new DO-file to be activated. This function may be called with DE=0 to cancel all active and stacked DO-files.

Function 99: Disable/Enable Autoload

Called with: C = 99  
E = 0 to disable autoload  
-1 to enable autoload

Function 100: Send Command Line

Called with: C = 100  
DE = buffer address (or 0)

**Notes:** The first byte of the buffer must contain the command line length, with the command line text starting at the second byte of the buffer. This function causes any currently-active command line to be stacked, and the new command line to be activated. This function may be called with DE=0 to cancel all active and stacked command lines.

**Function 101: Set/Return Spool Mode**

**Called with:** C = 101  
D = printer assignment (if spool mode = 0)  
queue assignment (if spool mode = 1)  
-1 to leave assignment unchanged  
E = spool mode:  
0 to print direct  
1 to print spooled  
2 to print to the console  
bit 7 = 1 for remote, 0 for local  
-1 to leave spool mode unchanged

**Returns with:** A = 0 if successful  
-1 if invalid parameters  
H = current printer or queue assignment  
L = current spool mode

**Notes:** Printer and queue assignments are coded 1="A", 2="B", ..., 16="P". Assignment to queue zero causes print files to be left unqueued. Assignment to printer zero causes print output to be discarded. Setting the assignment or mode causes any currently-open print file to be closed and processed in accordance with the previous assignment and mode. If both D- and E-registers are -1, then this function returns the current assignment and mode in the HL-registers.

**Function 102: Set/Return Spool Drive**

**Called with:** C = 102  
E = spool drive (0="A", 1="B", ..., 15="P")

**Notes:** Setting the spool drive causes any currently-open print file to be closed and processed in accordance with the current spool mode.

**Function 103: Set/Return De-Spool Mode**

**Called with:**

- C = 103
- B = printer (0="A", 1="B", ..., 15="P")  
bit 7 = 1 for remote, 0 for local
- D = de-spool queue assignments:  
1="A", 2="B", ..., 16="P"  
0 to set printer offline  
-1 to leave queue assignment unchanged
- E = de-spool mode:  
0 to process print job  
1 to suspend print job  
2 to restart print job from the beginning  
3 to terminate print job  
-1 to leave de-spool mode unchanged

**Returns with:**

- A = 0 if successful  
-1 if invalid
- H = current queue assignment 0...16
- L = current de-spool mode (1 if stopped, 0 otherwise)

**Notes:** If both D- and E-registers are -1, then this function returns the current assignment and mode in the HL-registers.

**Function 104: Queue Print File**

**Called with:**

- C = 104
- DE = FCB address
- B = queue (0="A", 1="B", ..., 15="P") plus  
bit 7 = 1 if remote, 0 if local  
bit 6 = 1 if queued files to be deleted after printing

**Notes:** Only the first 12 bytes of the FCB (Drive, Name and Type) are used, and no validation is performed.

**User's Guide to TurboDOS**  
Copyright (C) 1981 by Software 2000 Inc.  
Additional Functions

**Function 119: Send to Network**

**Called with:** C = 119  
DE = message buffer address

**Returns with:** A = 0 if successful  
-1 if unsuccessful

**Notes:** Do Not Use.

**Function 120: Receive from Network**

**Called with:** C = 120  
DE = message buffer address

**Returns with:** A = 0 if message received  
1 if negative acknowledgement received  
-1 if unsuccessful

**Notes:** Do Not Use.

**Function 121: Allocate Memory Segment**

**Called with:** C = 121  
DE = length of segment requested

**Returns with:** HL = address of segment  
A = 0 if successful  
-1 if insufficient memory

**Notes:** Do Not Use.

**Function 122: De-Allocate Memory Segment**

**Called with:** C = 122  
DE = address of segment

**Notes:** Do Not Use.

**Function 123: Send Inter-Process Message**

**Called with:** C = 123  
DE = address of node name  
HL = address of message

**Notes:** Do Not Use.

**Function 124: Receive Inter-Process Message**

**Called with:** C = 124  
DE = address of node name

**Returns with:** HL = address of message

**Notes:** Do Not Use.

**Function 125: Delay Process**

**Called with:** C = 125  
DE = delay count (system ticks) or 0

**Notes:** Do Not Use.

**Function 126: Create Process**

**Called with:** C = 126  
DE = address of process entrypoint  
HL = address of process work area

**Returns with:** HL = address of new process descriptor  
A = 0 if successful  
-1 if insufficient memory

**Notes:** Do Not Use.

**Function 127: Terminate Process**

**Called with:** C = 127  
DE = address of process process descriptor (or 0)

**Notes:** Do Not Use.



# SOFTWARE 2000 inc.

P.O. Box 946 • Los Alamitos, California 90720 • 213/429-2317

---

