Preface

TDOS Manual Update

This manual is an edited and updated version of the original manual created by AJ Morehen and Guy Gousineau. The majority of the information has not been changed except for spelling and formatting errors. Additional information has been added concerning various utilities that are included with TDOS.

William (Milli) Hicks
http://smartbasic.net – 2017

TDOS SYSTEM DOCUMENTATION
By AJ Morehen & Guy Cousineau

TDOS is a public domain CP/M replacement operating system. It may be distributed freely but under no condition shall it be sold without the permission of the author.

This user manual is similar to TDOS.HLP in that it contains the same information. It is already pre-formatted for LISTing to your printer via a "LIST TDOS.DOC N" command.

It will guide the new or experienced user through the features of the TDOS enhancements to the CP/M system for the COLECO ADAM computer.

DO NOT make any changes to TDOS.HLP as its format is critical to the operation of TDOSHELP.COM.

TDOS Manual Structure

This manual is comprised of three major sections; user documentation, utilities, and installation. The entire Adam community is deeply indebted to Tony Morehen and Guy Cousineau for their substantial and continuing efforts to provide a well-documented, robust operating system complete with utilities for our Adams.

Contents

WHAT IS TDOS?

TDOS is an alternative operating system for the Coleco Adam that is 99% compatible with Coleco's CP/M 2.2. CP/M stands for Control Program Monitor. TDOS, like CP/M, helps the different parts of your ADAM communicate with each other so you to use them to greater advantage. You can run a whole new series of programs that can make it into an efficient, powerful home computer. This new operating system will provide you with more capabilities and features than the EOS or standard CP/M systems. The best part is that it offers you all of these enhancements free of charge! Add to that the ability of TDOS to access hardware and software previously to be inaccessible to Adam owners.

TDOS has corrected several deficiencies and bugs in the original CP/M system as well as introducing advanced features to help you get the most of the operating system.

TDOS is a very flexible system which can handle disk drives of various sizes, ram disks up to 1 megabyte, parallel printers, internal and external modems, and much more.

The TDOS system can support 2 types of hard drive conversions. One of the conversions (the BJ controller) involves taking your Adam apart, desoldering your CPU chip, and adding a circuit board inside your ADAM. The other (the Micro Innovations controller) is a plug-in version which goes in the center expansion slot.

With either system, you can plug in a 10, 20, or 40 megabyte hard disk drive. Each drive can be partitioned for a maximum of 32 MB of storage under TDOS. In addition, any portion of the drive may be reserved for use under the EOS operating system (SmartBASIC, SmartWRITER, etc).

TDOS is smart enough (via the user installation) to access only the devices that you have at the time of installation. If you change your system configuration by adding a disk drive or other peripheral, you must re-install TDOS.

TDOS comes with many built-in features which reduce the number of utilities required to run the system. See resident commands for more information.

It also comes with its own set of specific utilities to help manage the system and files. See the Utilities topics for more information.

Throughout all its development, TDOS has been kept compatible with CP/M 2.2. This means that most programs written for CP/M 2.2 should run on TDOS without modification.

Beware of software from disreputable sources. The developers of TDOS will assume no responsibility for data loss or damage resulting from the use of defective software, including our own.

STARTING TDOS

To start TDOS, insert the disk on which you installed your system into disk drive 1 and pull the reset switch. There is disk activity and the system loads itself into memory. In inverse video, at the top left of the screen, you will see TDOS followed by the current revision number. Below this, you will see the TDOS prompt.

At the same time that TDOS is loaded into memory, the CCP and BDOS are also copied to an unused area in the Video Ram so that when programs exit with a warm boot, the system is instantly re-copied to memory. You will no longer have to wait several seconds for reboots when exiting from your favourite programs. Note that if video memory becomes corrupted, TDOS will tell you that the system is invalid and ask you to put in a boot disk. Put a system disk in the original boot drive and press return. Hard disk users should simply press return.

NOTE: If your keyboard is disconnected, TDOS will let you know by printing a string of -@-@-@ across the screen.

The TDOS prompt: AO> lets you know that it is waiting for your command. The letter A lets you know that all commands will be directed to Drive A, the drive from which you booted your TDOS while the 0 lets you know that all commands will be directed to user area O.

From this point on, TDOS has control of the computer and will attempt to execute any task you tell it.

OPERATING SYSTEM

An Operating System is what runs your computer. An OPERATING SYSTEM (OS), is a collection of machine-language routines that control the computer, run the external devices (such as printers, modems, disk drives, etc.) and accesses the keyboard for input and control information. In other words, it sees what you tell it to see, and then does what you tell it to do. Most importantly, the operating system provides the file and disk management functions that allow you to conveniently and quickly save your programs and data files for future use.

EOS is the default operating system that comes with the ADAM while CP/M 2.2 and now TDOS are systems that replace EOS with a brand new OS that stay there until you clear them out by pulling the reset switch or by powering down.

One question that is often raised by EOS users is why should I switch to CP/M or TDOS? The answer is quite simple. TDOS offers more power, greater flexibility and access to thousands of commercial and public domain software programs. Many of the most well known software companies had their start in CP/M: Wordstar for word processing, Dbase II for databases and Supercalc for spreadsheets. These applications overshadow Smartwriter and AdamCalc. Furthermore, TDOS lets you expand the Adam to its fullest potential. 80 column displays, high speed modems, and hard drives are fully supported by TDOS and by all programs running under TDOS.

CONSOLE COMMAND PROCESSOR (CCP)

The CCP, BDOS, and BIOS are the 3 components of the operating system. See the sections devoted to the BDOS and BIOS for more information. The CCP is the interface between the operating system and the user. It is through the CCP that you send commands to the operating system that tell it to run an application program or to copy a file from one disk or user area to another.

CP/M and TDOS use what is known as a command line interface to interact with the user. MS-DOS uses a very similar interface. Indeed, once you get used to TDOS, there is almost no effort involved in learning MS-DOS.

When using a command line interface, you simply wait for the operating system to display its command prompt and then type in a command that can be a simple one word or a very complex command. For example, to use the Wordstar word processing program, you would enter:

A0>WS<ret>

A more complex command would be:

A0>TYPE TDOS.DOC /N<ret>

This would tell the operating system to display the file that you are now reading on the screen, without pausing when it fills up the screen (TDOS normally waits at the end of every screen for you to press a key in order to continue). This is about as complex as most TDOS commands get, however, some programs can require exceptionally complex command lines (those programs usually also offer a simpler but slower menu driven interface).

EOS, on the other hand, does not use a command line interface. In fact, it could be argued that EOS does not have any interface at all! After all, all you do is put in a disk and pull the reset switch. While the EOS approach is very simple, the command line approach of TDOS is not much more complicated and is much more flexible.

For example, to run AdamCALC under EOS, you first put your AdamCALC disk in one of your drives and pull the reset switch. Simple and straightforward. But then so is the TDOS approach. To run SuperCalc (a spreadsheet program that is much better than AdamCALC), simply put your Supercalc disk in drive A: and type SC<ret> and away you go!

However, suppose you wanted to copy a SmartBasic file from one disk to another, how would you do it? One method would be to run SmartBasic, load the file into memory from the source disk, put the destination disk in a drive and save the file to the destination disk. Very complicated. Alternatively, you could boot a file copy program and use it to copy the file. Obviously, EOS makes it very difficult for you to carry on such basic file management tasks as copying, renaming or deleting files.

TDOS makes those tasks very easy. For example, to copy an MBASIC file from drive A: to drive B:, you would enter at the AO> prompt:

A0>COPY A:ANYFILE.BAS B:<ret>

and TDOS would do it all for you. There is no searching for the disk with the copy utility or no long pauses waiting for SmartBasic to load. Deleting or renaming files is just as easy. For more information on TDOS's built-in file maintenance commands, see the RESIDENT COMMANDS section.

However, there is one problem with a command line interface: if you make a mistake, you have to re-enter the entire command over again, risking another mistake. TDOS minimizes this problem by maintaining a command history that lets you recall previous commands and improved editing features that let you correct your mistakes. For more details on these features, see the COMMAND HISTORY/LINE EDITING section below.

Another advanced feature of TDOS is that it lets you enter more than one command per line. All you do is separate each command with a semi-colon: ";". This significantly speeds up the response to the second and subsequent commands. Note that one command line cannot exceed 200 characters.

In summary, the CCP is the frontline interface between you and your computer. At your command, it performs basic file management tasks with its built-in commands as well as loading and running application programs. Finally, the CCP is responsible for processing your batch files (see that subject) which let you automate some of the more repetitive computer operations.

BASIC DISK OPERATING SYSTEM (BDOS)

The BDOS is responsible for character input output functions, and is a major player in file input and output. Most programs use BDOS functions to accomplish their major input output tasks.

CP/M had a protection feature that was supposed to prevent accidently writing to a disk unless it was logged in with a ^C or disk reset. This was one of the most annoying features of CP/M that often caused data to be lost because a disk was full. The TDOS BDOS detects changed disks and automatically logs them in. There is no need to warm boot the system each time you change disks.

BASIC INPUT OUTPUT SYSTEM (BIOS)

The BIOS is the high level interface between the programs (and BDOS) and the physical devices composing your system. It has the necessary drivers to access your TV or monitor, the tape and disk drives, the expansion memory, hard drive, etc.

The TDOS BIOS supports up to 4 disk drives of different sizes, and one hard disk partitioned into 4 logical drives. Drives are re-ordered in order to improve system performance. The faster drives are placed before the slower ones. The TDOS installation program will tell you about the actual order.

DRIVES AND USERS

CP/M is a disk based system. File and Program operation is accomplished by LOGGING into a drive and accessing the files on it. Since the BDOS and BIOS together handle the different access methods for various types of drives (disk, tape, ram disk, hard disk), the programmer only needs to know the logical label assigned to them.

When you installed you TDOS system, you were given a summary of your Drive allocations. It might have looked like:

A Hard Drive Volume 1
B Hard Drive Volume 2
C Hard Drive Volume 3
D Hard Drive Volume 4
E Ram Disk
F Disk Drive 1
G Disk Drive 2
H Tape Drive 1

Under TDOS, the hard drive is partitioned into 4 logical drives: A, B, C, and D.

Since drive A is recommended as a UTILITY drive, it will usually be smaller than the other 3. The maximum drive size is currently 8 MEGABYTES per drive.

A typical floppy disk system will look like:

A Ram Disk
B Disk Drive 1
C Disk Drive 2
D Tape Drive 1

In order to LOG into disk drive 1, simply type: B:<ret>. The system prompt will change to:

B0>

The 0 indicates that user 0 has been selected; this is the default. There are 32 user areas allocated to each drive to allow you to organize your files in the equivalent of sub directories. On a 160K (single sided) disk, this may seem trivial. But when you get into hard disks with 1,000K or 7,000K of storage, you can appreciate the need for user areas.

All files on a medium are placed in the same directory area, but there is a tag in the first byte to indicate which user area a file belongs in. Disk activity will take place on files in the current user area. To change user areas, you can just type the user number followed by a colon:

B0>3:<cr>
B3>a2:<cr>
A2>

In the second example, notice that a drive and user have been specified together.

This method of changing users replaces the CP/M 2.2 USER command.

See the CD command for more on users.

FILE NAMES

A filename has 3 components:

1) DRIVE/USER - [du:]

The drive designation is a letter ranging from A to P identifying the drive on which you expect to find the file. In addition, some applications programs let you specify the user number of the selected drive. See drives and users for more information. A drive name is followed by a colon:

A:
3:
B6:

When program syntax calls for [d:] or [du:], it means that you can optionally specify a drive or a drive/user for your file.

TDOS, of course, supports full use of the [du:] specification for its own internal use.

2) FILE NAME - nnnnnnnn

The file name is up to of 8 alphanumeric characters. In addition, TDOS allows the use of special characters in a file name except for the following:

. , ; < >

Thus valid file names could be something like:

GAMES MY_PROG SWAP21 DUMP+ TDOSHELP FINDREPL

With a bit of practice and innovation, you can usually find a descriptive 8 character name for your files.

3) Type - ttt

The file type is 3 characters and denotes the nature of the file. Some standard file types are:

ASM 8080 assembly code
BAK Editor Back up file

BAS	Basic programs
COM	Executable command file
DOC	Documentation file
HEX	Intermediate assembler file (usually Intel Hex format)
LST	Assembly listing
PRN	Assembly listing
REL	Intermediate assembler file
SUB	Submit/Batch file
SYM	Assembly symbol table
TXT	Text file
Z80	Z-80 assembly code (source code file)

The 3 components of a file name are put together in the following fashion:

[du:] nnnnnnnn.ttt

Examples:

A:MY PROG.COM A3:MY PROG.DOC 3:MY PROG.BAK MY PROG

Note that there is always a COLON after the drive/user specification and that there is a period between the file name and the file type. If there is no file type, a period may be used to indicate the blank field. If there is no drive specification, the operation is performed on the current drive and user area (default drive).

Standard program syntax (usually found in the DOC file) will usually explain how to specify a file name. The directory program, for example has the following syntax:

DIR [du:][afn]

The square brackets denote optional values. AFN denotes an ambiguous file name (with wild cards). Alternately, programs which do not allow wild cards will use UFN to specify an UNambiguous file name.

Wild cards are useful to find several similar files, or to tag several files for multiple operations. There are 2 types of wild cards:

The ? matches any character in a particular position in the file name or file type

Thus MY_PROG.?O? would match MY_PROG.COM and MY_PROG.DOC; it would not match MY_PROG.BAK since there is no '0' in the second position of the file type.

The * wild card will match any character in the remainder of the file name or file type. For example:

DIR MY PROG.* would match:

MY PROG.COM
MY PROG.DOC
MY PROG.BAK

Once a * wild card is used, any subsequent characters will be ignored. Similarily, if you type more than 8 characters in the filename or 3 in the filetype, the extra will be ignored. Following are some valid and improper file names for the directory program:

VALID	IMPROPER	Reason
A:MY?PROGR.*	A:MY?PROGRA.*	(name too long)
MY-)")"G D*	MY7??w7G D*C	(no character
B6:M*.COM	B6:M*G.COM	allowed after *)
M*.?0?	M*.:0?	(illegal character)

Note that the directory will always store a file name as 8 characters and a file type as 3. A file could be named just "A.", but the directory will report it as "A . ", filling in the unused characters with spaces.

RESIDENT COMMANDS

Resident commands are so called because they reside in the CCP and are available for immediate execution. If you type:

DIR<ret>

For example, the CCP finds that this is a resident command and executes it.

If you gave a program the same name (DIR.COM), the CCP will refuse to execute it since the name matches the resident DIR command. You can override this feature by specifying a drive name on the command line:

A:DIR<ret>

DIRECTORY

The resident directory program will report files in any drive/user area; it is not necessary to log into a drive in order to get a directory listing. the DIR command has the following syntax:

DIR [du:][afn]

This means that you have the option of specifying a drive, a user, a drive/user combination, and ambiguous file name (with wild cards) or an unambiguous file name.

If no file name is specified, the DIR command will list all files.

These are all valid forms of the directory command:

DIR
DIR *.COM
DIR READ.ME
DIR A:
DIR A:*.?O?
DIR B5:
DIR 3:TDOS.COM

The directory will show the size (in K) for each file, report on the number of files matched, the total size of all matched files, the storage space on the drive, and the space remaining on the drive.

DIR will pause after it fills each screen. At this point press any key to continue or ^C to abort.

TYPE

The type command will type a file to the screen 24 lines at a time, pausing at each screen. Simply press any key to carry on. If you want to abort the display, press -C when the display pauses.

TYPE does not check the TYPE of file that has been requested. DO NOT type a .COM file or any other file that contains non-ASCII characters. Although no damage usually results from such actions, you might not be too pleased with the results.

Similar to the DIR command, TYPE can type a file from any drive/user area. It has the following syntax:

TYPE [du:]UFN [/n]

Note that no wild cards are allowed in the file name as denoted by the UNAMBIGUOUS file name.

The [/n] option lets you disable the paging. If you want to type the TDOS.DOC file without any screen pauses, just enter:

TYPE TDOS.DOC /N

LIST

List is similar to TYPE but it sends the selected file to the screen and printer. It adjusts it's default page length to 58 lines and sends a FORM FEED to the printer at every page.

If you are using the ADAM printer which does not interpret FORM FEED, you don't have to worry: the BIOS will simply wait for you to change the paper or advance the paper to the next page. Once your next page is ready, press RETURN to proceed.

If you want to LIST a file that already has form feeds in it, use the '/n' option to suppress TDOS's own pagination.

COPY

The COPY.COM utility that comes with CP/M had difficulty copying small or large files. PIP.COM also had problems with certain files. TDOS has its own resident command which will let you copy files from one drive to another, from one user area to another and even within the same user area. Its' syntax is:

COPY [du:]Source_File_Name [du:]Destination_File_Name, or

COPY [du:]afn [du:], or

COPY [du:]ufn [du:]ufn

(at least one [du:] must be given)

A few examples will help illustrate the versatility of this command:

CO> COPY *.COM B:
Copy all the .COM files from drive CO (the default drive) to drive B (user 0 is assumed).

C5> COPY A:*.COM
Copy all the .COM files from drive A (user 5 is assumed) to drive C5 (the default drive).

C5> COPY *.COM 6:
Copy all .COM files from C5 to C6.

C5> COPY AO:B*.COM B2:
Copy all .COM files starting with 'B' from AO to B2.

C5> COPY HELP.COM B:
Copy a single file from C5 to B5.

C5> COPY 6:HELP.COM
Copy single file from C6 to C5.

C5> COPY 6:HELP.COM 5:NEWHELP.COM
Copy HELP.COM from C6 and give it the new name NEWHELP.COM in C5 .

C5> COPY HELP.COM OLDHELP.COM
Make a copy of HELP.COM in this drive/user and give it the new name of OLDHELP.COM

Note in all cases that the original file is unaffected by the copy operations. If a new file name is specified on a wild card operation, it is simply ignored.

DELETE

CP/M used the ERA command to delete (erase) files; TDOS prefers DEL. The DEL command allows you to delete any file or group of files from any drive/user area. Its' syntax is:

DEL [du:]afn

Since delete removes a file from the directory, it should be used with care. Valuable programs can be lost by reckless use of the DEL command.

As a safety feature, DEL will ask for confirmation if you ask to delete all files with a "DEL *.*".

RENAME

The REN command will change the 'name of a file. The file itself is not copied or modified, just the directory entry.

CP/M syntax for renaming files insisted that you specify the new name first, use an '=' and follow with the old name. TDOS uses the more logical approach of renaming from the old name to the new name without the =.

```
REN [du:]ufnl ufn2
```

Where ufnl is the original name and ufn2 is the new name. It is not necessary to specify a 'du' for the new name; TDOS will ignore it since REN does not copy.

CLEAR SCREEN

On occasion, you may wish to clear the screen. Simply type CLS and it is done instantly.

SAVE

SAVE is an advanced command which allows you to capture what currently resides in memory starting at 100H. The Syntax is:

SAVE nn [du:]ufn

where nn is a decimal number representing the number of 256-byte pages to be saved to the designated file name.

All SAVE commands start to capture memory at 100H, the beginning of the transient program area (TPA).

GO

On occasion, you may accidently exit from a program only to find that you want to execute it again. It may take several seconds to reload the program from disk.

The GO command allows you to immediately re-enter the program currently in memory.

Some programs, especially the ones with overlays or self modifying code cannot be safely re-entered in this fashion. These programs are usually protected and the GO command has no effect.

When using the GO command, you can optionally pass parameters to the program in memory just as if it had been loaded from disk. For example, suppose you have forgotten the syntax for the PATH command and ask for help with: AO>PATH ?

After consulting the help message, you can now change the path by typing:

GO A0<ret>

to instantly change the path without waiting for the program to load again from disk.

TRANSIENT COMMANDS

Transient commands (programs) do not reside in memory; they have to be loaded from disk every time they are executed. To run a program in TDOS you use files that have a file type of COM. Type in their name as it appears in the directory without the ".COM". Program commands can be grouped together for batch-like processing in submit files, that is, files that have a ".SUB" extension. Submit files and command files can have the same filename. If so, the submit file will run before the command file. However, you can override this default processing order by typing the command filename with the ".COM" file extension. ie RUN.COM <command tail>.

Section 26 provides more information on submit files.

Any program that is designed to run on a CP/M 2.2 system should run on TDOS. There are thousands of programs available for the CP/M environment ranging from utilities to text processors, games, data base management, and much more.

A few transient programs are included with TDOS; see the help topic on utilities for descriptions of these programs.

Most programs will come with their own documentation, included in a progname.DOC file or in a READ.ME file. Please read these carefully prior to using any new program on your system. Some programs may need to be installed so that the program knows the type of system you have. Instructions on installation are usually provided in the documentation file.

BEWARE of software from unreliable sources: they may do irreparable damage to your system.

UTILITIES

TDOS comes with its own set of utilities. These can be used to accomplish certain maintenance activities.

Refer to the help menu for a list of these utilities.

The TDOS system and its utilities should replace most of the standard CP/M utilities like:

STAT.COM	COPY.COM
SYSGEN.COM	SUBMIT.COM
ADAM.COM	CPMADAM.COM
CONFIG.COM	BACKUP.COM

If you already have COLECO'S CP/M system, DO NOT use ADAM.COM, CPMADAM.COM BACKUP.COM, or CONFIG.COM. They address the operating system in a non-standard way and could crash the system or destroy your disks.

With the implementation of TDOS, you no longer need to sysgen all your disks. A simple format is all you need to prepare your disks for use in CP/M.

BACKUP was used to perform an image copy of a CP/M disk including the system tracks. This is also not required in TDOS. To make a back up, simply copy all the files from one disk to another.

CD - CHANGE DIRECTORY

TDOS CD program is used to assign names to user areas and change between user areas using logical names. If you do not have a hard drive, this utility is of limited use.

Typing CD alone will give a short help message and follow with a list of all the defined user areas in your system.

Besides changing to your named directory, CD.COM can also MAKE or REMOVE directory names

CD.COM should be placed in drive AO where it can be accessed by the search path.

To create a directory name, first log into the drive and user that you want to name then type:
B6>cd <name> /n

where <name> is up to 8 characters defining the user area.

Use names like GAMES, DBASE, ASM, COMM, or any other name that describes the type of files in the current area.

CD will create a label name for the current area. The next time you log into that drive/user, the CCP prompt will report the logical name assigned to the drive/user.

To remove a label, you use the same syntax as create except use /r at the end of the command.

Once all your directories are named or if you change a directory name, log back into drive AO and type:

AO>CD /C

CD will ask which drives you wish to scan. It will then pick up all the label names and store them within itself. From that point on, to change directory, just type:

CD <name>

This will make it much easier to switch between user areas. Every time you change/add user area names, don't forget to CONFIGURE the CD command to update the list.

CU - CHANGE USER

Occasionally, you may wish to move a file from one user to another. While this may be accomplished by use of the COPY and DEL commands, the CU command will move the file without making another copy.

syntax: CU [du:]afn u:[/n]

You can specify a file or a group of files on another drive/user and provide its new user area. The program will prompt you before each change to be sure you really want to move the file.

If you forget the syntax, just type CU for a brief description.

The optional /n parameter suppresses the confirm prompt.

DATE - DATE STAMPING

The date stamping function is used by the BDOS to record the creation and modification date of each file. This can be handy when checking backups or revisions to files.

First, use the INITDIR to prepare your directories for time and date stamps. Type INITDIR D: where D is the drive to initialize. You must specify the drive. INITDIR will work even if you have files in the directory. It will not damage any files already on the disk.

If you have a clock/calendar, everything is now handled automatically. If not, use the DATE program to set the date when you turn on your system. This command can be put in your PROFILE.SUB (see submit).

In order to see the file dates, use the TDIR program. It uses the same syntax as the DIR command and will show files one line at a time with their creation and modification dates.

IOBYTE - INPUT/OUTPUT BYTE

The I/O byte is a control byte located in zero page. It consists of four 2-bit codes which map the architecture of your system.

They are preset at installation to default values for keyboard, screen, printer, and external devices.

The first 2 bits are for your screen (CONsole) and represent the keyboard and screen. The values are

00 CRT 40 column TV or monitor (80 column monitor if 80TDOSxx)
01 SR1 serial port 1
02 SR2 serial port 2
03 UC1 80 column monitor

Whether in 40 or 80 column mode, the default I/O byte for CON is 00

The next 2 bits are for the KEYboard.
00 KYB Adam keyboard
01 SR1 serial port 1

02 SR1 serial port 2
03 UK2 80 column terminal keyboard

The next bit is for the READER.
00 SR1 serial port 1
01 SR2 serial port 2

The next bit is the PUNCH device.
00 SR1 serial port 1
01 SR2 serial port 2

As these functions are seldom used, their setting is relatively unimportant.

The last 2 bits are for the LIST device or printer. The default values are:
00 LPT Adam Printer
01 SR1 serial port 1
02 SR2 serial port 2
03 PAR Parallel Printer

If you install a parallel printer, or wish to change back and forth between the ADAM and parallel printer, this byte must be changed.

When you install your system, you will be prompted to set the default values of your I/O byte. If you want to make a permanent change, you must re-install TDOS.

To make a temporary change, use the IOBYTE.COM on this disk. It will present you with the same menu and allow you to make changes that will remain in effect for the duration of the session.

PATH

When you type a command at the console, the CCP first checks to see if it is a resident command. Next it checks whether the file exists in the current drive/user area.

If the file is not found, the CCP uses an internal search path to look elsewhere for the file. The default path is:

'A drive user 0'

If the file is not found after the path search, an appropriate error message is returned.

In order to modify the path, use PATH.COM. If you just type PATH, the current path is shown for confirmation:

Path = $0 AO

The $ indicates CURRENT DRIVE.

To turn the path on or off, simply type:

PATH ON<ret> or PATH OFF<ret>

If you want to change the path, simply fill in path addresses (up to 5) on the command line:

PATH $0 B4 BO A0<ret>

would set the search path to

current drive user 0
drive B user 4
drive B user 0
drive A user 0

If you forget the path syntax, type PATH ? for a summary.

The path is most handy on a hard drive system. We recommend placing your frequently used utilities in drive AO. This way you can have instant access to them.

Additionally, if your drive is partitioned into related utilities, you may place utilities often used in drive B in BO, and so on with drives C and D.

Note that when searching for a file, whether in the current drive/user or on the path, TDOS will first look for files with a slow, even on hard drive systems. You can override the double search by specifying .COM for command files. Also, specifying the drive stops all path searches. Therefore, the fastest way to access a command file in drive A, user 0 is to enter the following:

```
B6>a0:tdir.com dl:
```

SUBMIT OR BATCH PROCESSING

TDOS has the ability to perform several operations from an instruction file called a SUBMIT file. A sample submit file would look something like:

```
DIR
1:
DIR 0:*.COM
```

A submit consists of one or more commands that you would normally type at the keyboard. Any legal TDOS command is permitted. However, it one submit file calls another, all submit processing will terminate after the second batch file is finished, even if there are more commands in the first submit file. This means that submit files cannot be nested.

When a submit file runs, TDOS does not display the commands in the submit file as it executes them. To see those commands, include an "ECHO ON" command in your submit file. "ECHO OFF" will turn the command display off again. "ECHO message" will display the message on the screen even if echo is off.

Create a sample BATCH.SUB file using your favorite text editor to enter the above commands then simply type BATCH at the command prompt. Then sit back and watch it perform all the specified instructions: first it gives you a directory of all files on drive AO>. Then it logs in to user 1 and shows a directory of all the .COM files in user O. Then the batch file terminates. Remember to type 0:<ret> to get back to user O.

You can also pass parameters to a SUBMIT file by including a '%' followed by a number from 1 to 9 in the file itself. Let's take an assembly session for example. You start by editing your source code file, then you assemble it, then you test run your program.

You can create a ASEM.SUB file that looks like this:

```
EDIT %1.Z80
ASM %1
%1
```

where EDIT and ASM are the names of your text editor and assembler. In order to edit and assemble GAMES.Z80, you would type:

A0>ASEM GAMES<ret>

The Batch file then takes over and executes the first command. When it gets to %1, it looks for the first parameter that you typed on the command line (GAMES) and makes the substitution.

Once the editing session is over, the batch file again takes over and performs the same substitution for the ASM command.

The final step is to run the command file.

If for any reason you want to abort a SUBMIT operation, just press any key while the system is rebooting between commands.

You can pass up to 9 parameters in the submit file and reference them with %1 %2 %3 etc. In addition, using %0 will pass the FILE NAME (and drive/user if specified but not the ".SUB" file type) of the submit file to the submit file itself. The submit file need not reside on the same drive/user as you are currently working on. Thus:

B6>A0:ASEM GAMES<ret>

Would run the submit file from drive AO while working on GAMES.Z80 in B6.

The cold boot sequence will also look for a PROFILE.SUB on drive AO. This can be handy to execute any task you want when you turn your system on.

Note that submit files run before command files. This can cause problems if the submit file tries to run a command file of the same name. If the submit file contains only the name portion of the command file, it will actually run itself again and again. To avoid this, always include the filetype (.COM) of command files.

Submit files can be speeded up by using TDOS's multiple commands per line feature. The multiple commands avoids excessive disk access reading the submit file. First, debug the submit file using one command per line and ECHO ON. Then combine the commands as much as possible separating each command with a semi-colon. Note that each line should not exceed 200 characters. Once this is done, the submit file will be much faster.

COMMAND LINE HISTORY AND EDITING

All the commands that you type at the CCP prompt are stored in a history buffer. In addition, any program that uses the READ CONSOLE BUFFER function will also have access to previous commands.

The number of commands that can be recalled is limited only by the size of the command buffer which is 256 bytes.

To recall a command, press the up arrow key. You can continue to press the up arrow key to recall earlier commands. To scroll forward through the list of commands, press the down arrow key. Only commands that are two or more characters in length are saved in the command buffer.

The current or a recalled command can be edited and re-executed. This is particularly useful for correcting syntax or other command errors or for repeating one command several times. Default CP/M systems use limited editing characters which were intended for hardcopy terminals. TDOS has implemented more practical line editing sequences.

You can scroll back and forth through a command line using the left and right arrow keys. Pressing HOME or HOME-RIGHT will take you to the start or end of the line.

Pressing DELETE will delete the character under the cursor; BACKSPACE will delete the character to the left of the cursor.

Pressing CLEAR will delete from the cursor to the end of the line. Pressing ESCAPE will delete the entire line.

Pressing INSERT will toggle between INSERT and OVERSTRIKE mode. This is useful when correcting typing errors where a character was omitted.

The CCP uses the console buffer for entering your commands and all line editing characters will function. In addition, applications programs using the console buffer function will let you edit your commands. If

the editing commands do not function, the program is likely not using the command buffer to get your input.

Since the command line editing characters have been assigned to certain special keys on your keyboard, you should not redefine these keys (see keyboard translations) unless you are prepared to use the alternative control keys:

CLEAR	^Y
INSERT	^V
DELETE	^G
BACKSPACE	^DELETE
ESCAPE	none
HOME	^Q
HOME-RIGHT	^F

TDOS PROGRAMMING

For the programmer, the BDOS has a standard syntax or calling convention. Load the C register with the function to be accomplished. If the function requires a parameter, it is passed in register DE. DOSCALLS.LIB, included with TDOS, can be used by programmers. It contains equates and a summary description of all TDOS functions, including those unique to TDOS.

BDOS Functions

FUNC	NAME	PARAMETERS	RETURN CODE
0	System Reset (Warm Boot)	none	none
1	Console Input	none	character in A
2	Console Output	E = character	None
3	Reader Input	none	character in A
4	Punch Output	E = character	none
5	Printer Output	E = character	none
6	Direct Console I/O	E = 0FFH	character in A
		E = 0FEH	console status in AF
		E = 0 to OFDH	character out
7	Get I/O byte	none	I / O byte in A
8	Set I/O byte	E = I/O byte	none
9	Print String	DE = buffer address	none
10	Read Console Buffer	DE = buffer address	Characters in buffer
11	Get Console Status	none	Console status in AF
12	Get Version	none	HL = 22H
			DE = TDOS rev no.
13	Reset disk system	none	none
14	Select Disk	E= Disk Number	A = Directory code
15	Open File	DE = FCB address	A = Directory code
16	Close File	DE = FCB address	A = Directory code
17	Search First Directory	DE = FCB address	A = Directory code
18	Search Next match	none	A = Directory code
19	Delete File	DE = FCB address	A = Directory code
20	Read Sequential	DE = FCB address	A = Directory code
21	Write Sequential	DE = FCB address	A = Directory code
22	Make File	DE = FCB address	A = Directory code
23	Rename File	DE = FCB address	A = Directory code
24	Get Login Vector	none	HL = vector
25	Get Logged Disk Value	none	A = disk
26	Set DMA Address	DE = DMA	none
27	Get Allocation Vector	none	HL = pointer to ALLOC
28	Write Protect Disk	none	none
29	Get Read Only Vector	none	HL = vector
30	Set File Attributes	DE = FCB address	A = Directory code
31	Get Disk Parameter Block	none	HL = DPB Address
32	Set / Get User Code	E = 0FFH or user	A = User number (if 0FFH)
33	Read Random	DE = FCB address	A = Directory code
34	Write Random	DE = FCB address	A = Directory code
35	Compute File Size	DE = FCB address	Set in FCB
36	Set Random Record Number	DE = FCB address	Set in FCB
37	Reset Drive	DE = Login Vector	A = 0

FUNC	NAME	PARAMETERS	RETURN CODE
38*	Get DMS Address	none	HL = DMA address
39*	Change File's user #	DE = FCB address	Dest. User # in S1
41*	Get Time	DE = user buffer	time in user buffer
42*	Set Time	DE = user buffer	none
	For functions 41 and 42, the user buffer should be 6 bytes long and contain the time in BCD YY MM DD hh mm ss		
43*	Get File Stamp	none	none
44*	Use File Stamp	none	none
45*	Set / Get Program Code	E = 0FFH or code	A = Directory code
46*	Get Drive Label (name)	none	A = Directory code
47*	Set Drive Label (name)	DE = FCB (Name)	A = Directory code
48*	Get permanent media	none	HL = Permanent media bit-map
49*	Direct Bios Call	E = BIOS vector number see BIOSCALL.LIB HL = Parameter to be used by BIOS	none

DIRECTORY CODES for OPEN/CLOSE A=0,1,2,3 Directory offset, 0FFH=not found On READ/WRITE 0=successful non-zero=Read/Write error

Special Programming needs may require direct access to the BIOS. This is done through a series of jump vectors at the beginning of the BIOS. While many of them are the same functions used by the BDOS, there are some which cannot be accessed any other way.

Accessing the BIOS is a tedious operation which requires the handling of the warm boot vector in zero page:

```
LD      HL,(1)          ;get BIOS start
LD      L,OFFSET        ;Correct vector address
LD      BC,PARAMETER    ;add in parameters
CALL    JP HL           ;call a routine which jumps to (HL)
```

TDOS offers an alternate way of handling this sometimes tedious task via a BDOS function. The HL register is used to pass the parameters that the BIOS expects in BC, Register E has the BIOS function number (see BIOSCALL.LIB), and BDOS function 45 handles the rest. For example, to send a 'C' to the screen via a direct BIOS call, use:

```
LD      C,45            ;BIOS call
LD      L,'C'           ;character to send
```

```
LD      E,4             ;function number
CALL    5               ;BDOS does rest
```

Refer to the table below and BIOSCALL.LIB for BIOS functions

BIOS FUNCTIONS

FN	OFFSET	NAME	PARAMETERS
0	00	Cold Boot	none
1	03	Warm Boot	none
2	06	Console Status	none
3	09	Console Input	none
4	0C	Console Output	C = char (put in L for BDOS function 49)
5	0F	List Output	C = char (put in L for BDOS function 49)
6	12	Punch Output	C = char (put in L for BDOS function 49)
7	15	Reader In	none
8	18	Home Disk	none
9	1D	Select Disk	C = disk (put in L for BDOS function 49)
10	21	Select Track	BC = Track (put in L for BDOS function 49)
11	24	Select Sector	BC = Sector (put in L for BDOS function 49)
12	27	Select DMA	BC = DMA (put in L for BDOS function 49)
13	2A	Read Disk	none
14	2D	Write Disk	none
15	30	List Status	none
16	33	Sector Translate DE = Vector BC = Track	do not use BDOS function 49
	36 to 54 unused		
29	57	Smart Keys On	none
30	5A	Smart Keys Off	none
31	5D	Trap Smart Keys	none
32	60	Don't Trap Smart Keys	none
33	63	Key Display On	none
34	66	Key Display Off	none
35	69	Auxiliary Out	C = char (put in L for BDOS function 49)
36	6C	Auxiliary In	none
37	6F	Aux Out Status	none
38	72	Aux In Status	None
39	75	Bios Time	Time in (HL) 6 byte BCD – YY MM DD hh mm ss
40	78	Read One Block A = device DE = Buffer BC = Block Number	none
41	7B	Write One Block A = device DE = Buffer BC = Block Number	do not use BDOS function 49

FN	OFFSET	NAME	PARAMETERS
42	7E	AUX2 Out	C = char (put in L for BDOS function 49)
43	81	AUX2 In	none
44	84	AUX2 Out Status	none
45	87	AUX2 In Status	none
46	8A	Flush Console	none

ZERO PAGE USE

CP/M systems use the first page of memory from 0 to 256 for several buffers and pointers. In the following description of the zero page vectors and data areas, all values will be shown in HEX only.

00-02 JP WARMBOOT

This is the way that most programs exit back to the system via a JUMP 0 which in turn jumps back to the reboot code in the BIOS. This address is also used by programs to access BIOS functions (see BIOS) for a list. If you want to execute the third vector in the BIOS at offset 06, you could use the following code:

```
LD    HL,(1)
LD    L,06
JP    (HL)
```

The first instruction gets the page address of the BIOS in register H. The second instruction, replaces the WARM BOOT address with the address of the third function. The third instruction sends my program into the BIOS.

03 DEFAULT USER

Address 3 contains current user number. It is used by the BIOS to return your systen to the default user after a warm boot.

04 DEFAULT DRIVE

Address 4 contains the current drive. See DEFAULT USER.

05-07 JP BDOS

CALLing address 5 will execute a jump to the start of the BDOS and perform the selected function. Some programs use the address at bytes 6 and 7 to determine the top of memory. This is a good idea as it helps prevent system crashes.

08-3F	reserved
40-41	pointer to command history buffer
42-43	smart key return string
44-45	smart key description
46-47	pointer to search path

These 4 sets of bytes can be accessed by programs to temporarily redefine the smart keys, to access the search path, or to scan the command buffer.

48-5B	reserved
5C-6B	First File control block
6C-7B	Second File control block
7C	record number for first FCB
7D-7F	random record number

See FCB for further description of the use of the FCB.

80-FF	DMA buffer

This is the standard Direct Memory Address location. File access normally takes place in 128 byte sectors which are placed here and optionally moved to another location.

FILE CONTROL BLOCK

The file control block is a sequence of 36 bytes which contain information essential to file access. It has the following format:

```
Byte      000000000011111111112222222222333333
Number    012345678901234567890123456789012345

Data      DnnnnnnnnttttE12R0123456789ABCDEFR123

          ||        | ||  ||                  ||
          ||        | ||  ||                  |+---->  random record
          ||        | ||  ||                  +------>  current record
          ||        | ||  |+-------------------------->  allocations
          ||        | ||  +-------------------------->  record count
          ||        | |+------------------------------>  2 system control bytes
          ||        | +------------------------------->  extent number
          ||        +-------------------------------->  3 characters for file type
          |+-------------------------------------------->  8 characters for file name
          +-------------------------------------------->  drive number
```

The first byte of a 36 byte FCB is the drive code. Here 1 means drive A, 2 means drive B, and so on. 0 signifies the default drive which simply means the drive that you are currently logged into.

Bytes 1 through 8 are used for the primary file name, the part appearing before the '.1.

Bytes 9 through 11 hold the file type, a maximum of 3 characters.

Byte 12 holds the extent number. This indicates the number of 16K segments in a file. A 20K program would have the first 16K in extent 0 and the last 4K in extent 1.

Bytes 13 and 14 are system control bytes. The BDOS uses them for several checks and controls. For now let's just say that you must set them to 0 before opening a file...A good CCP (like TDOS) will do that for you but you must include it in your own programs.

Byte 15 holds the record count for the extent. Since there are 8 128-byte records per K, just knowing how many K there are in a file is not enough. This byte can be compared to the bytes-used-in-last-block of SMARTBASIC.

Bytes 16 to 31 hold the allocation numbers. Because CP/M treats disks as random access instead of sequential like SMARTBASIC, we must know where each block of a file is written on the disk.

Byte 32 holds the current record count. This is the record being written to or read from.

Bytes 33 to 35 are used with random files to determine the record number.

The first 32 bytes of the FCB correspond to a directory entry and are copied to the FCB when a file is opened. The other 4 bytes are manipulated by file access operations.

TDOS has 2 default partial file control blocks at 5CH and 6CH. These are only 16 bytes long for a total of 32 bytes. When the additional data is added too file control block number 1, it will overwrite the data in FCB2. It is the programmer's responsibility to move the data in the second one if it is required. Following the second FCB are 4 other bytes which provide the 36 full bytes required by FCB1. When the CCP interprets your commands, it places the next 2 'words' after the program (or command) name into FCB1 and FCB2. These words are treated as file names and are parsed accordingly.

ESCAPE SEQUENCES

Programmers may wish to control the screen and cursor directly in an applications program. Several escape sequences have been defined in TDOS. These are usually 2-character sequences with the first character being ESCAPE (27 decimal). The second character is an ASCII character; note that upper and lower case make a difference.

The escape sequences that follow are the ones supported by 40 column TDOS. These sequences are Heath H19 compatible, as are those used by the Eve VD-MB and Orphanware 80 column adapter. Other 80 column terminals may support different sequences.

ESC+ ACTION

A	cursor up
B	cursor down
C	cursor right
D	cursor left
E	clear screen
H	home cursor
I	reverse line feed
J	clear to end of screen
K	clear to end of line
L	scroll down entire screen by one line
M	scroll up entire screen
j	save cursor position
k	restore cursor position
l	delete line
o	clear to start of line
p	inverse on
q	inverse off

ESC Y is the introductory sequence for cursor placement. In order to position the cursor, 2 more bytes are required to represent the row and column respectively. 20H (32 decimal) must be added to the row and column. For example, to send the cursor to column 20 in row 5, you would send the following sequence:

27	Y	37	52
dec	ascii	dec	dec

For compatibility, ESC = is also accepted as an introductory sequence for cursor placement.

The following escape sequences apply only to the 40 column implementation of TDOS

ESC+ ACTION

m	scroll screen left 1 character
n	scroll screen right 1 character
i	nn sets the screen character color where nn contains the border (off) color in the high nibble and the set (on) color in the low nibble.

The following special sequences are all 3 characters. They start with ESCAPE, follow with x or y (in ASCII) and follow with an ASCII number. These sequences may not not work with your 80 column adapter, if installed.

ESC+ ACTION

x 5 cursor off
x 6 scroll off (40 column only)
x 7 don't trap smart keys
x 8 hide smart keys
x 9 smart keys off (show and trap)

y 5 cursor on
y 6 scroll on (40 column only)
y 7 trap smart keys
y 8 show smart keys
y 9 smart keys on (show and trap)

SMART KEYS

The 6 black keys at the top of the keyboard can return up to 15 characters including a <CR> if necessary. On the default system, these are configured to return the 6 most commonly used resident commands.

When you install TDOS, you will have the option of modifying these to your liking. We suggest that you leave them unchanged until you discover which commands would be most useful for you to have available with a single keystroke.

You can always reinstall TDOS later to change your smart key definitions.

The smart key installation consists of 2 sets of data for each string. The first is the display string which will show up at the bottom of your screen. It consists of 2 rows of 5 characters. Since these are entered on a single line, you will have to count the spaces in order to line up the second line of the display. The 80 column display will print the Smartkey descriptions in a single row on line 25 (if supported).

The second data element is the characters returned when that key is pressed. Since you may want to enter control characters in your string, only 2 editing keys are allowed. -X will erase the string so you can start over, and -z completes the entry of the string.

To turn the smart key display on or off, press SHIFT UNDO. Turning the keys off gives you 3 more lines of screen display on a TV screen. This may be essential in the running of some programs like full screen text editors.

If you want to turn the keys on and off under program control, refer to the escape sequences for further details.

KEYBOARD TRANSLATIONS

Keyboard translations are a means of interfacing your keyboard with the computer. They are used to return different values than are normally generated by the keyboard keys.

One example of keyboard translations are the smart keys which can be made to return a string of characters. Other keyboard translations are not as complex: they simply return a character that is different from the key pressed.

TDOS has already implemented WORDSTAR-like keyboard translations. The following list summarizes the major keys and their new values.

KEY	RETURN VALUE	KEY	RETURN VALUE
WILD	--	MOVE / COPY	^B
UNDO	^U	STORE / GET	^L
HOME-UP	^R	CLEAR	^Y
HOME-DOWN	^C	INSERT	^V
HOME-LEFT	^A	PRINT	^P
HOME-RIGHT	^F	DELETE	^G
HOME	^Q		

Certain keys perform specialized functions. ^C for example is the usual method of aborting a program. Note that some programs disable this function.

When a program writes to the screen and you want to pause the display, press -S to stop the screen; any other key will restart. Note that this function disables itself if other keys were pressed. When the screen is stopped, you can abort the program by pressing ^C.

In the 40 column mode, you can toggle the smart keys on and off with SHIFT- UNDO.

The ^P key acts as a printer switch when you are at the system prompt (A0>). Once you press ^P, all system output is echoed on the screen and printer. Pressing ^P again at the prompt will turn the printer off. To get a paper copy of the DIRECTORY for example:

A0>^PDIR<ret>

Press CONTROL-P (nothing happens immediately) and follow with DIR and a carriage return. The printer turns on and makes a copy of the directory as it appears on the screen. Once the printer is done, press -P again and it turns itself off.

The SHIFT-TAB key acts as a CAPS lock. After you press it, all characters from a to z will be converted to uppercase; this is much handier than using the LOCK key. To return to lower case, press SHIFT-TAB again.

If you are running a hard drive system, the SHIFT-WILDCARD will automatically park the heads. No more need for a PARK program.

In the 40 column mode, -LEFT and RIGHT will scroll the screen across 80 columns. This is handy when a program writes across the entire screen. Review the BDOS functions and use LEFT and RIGHT to see the whole screen.

DISKS

CP/M and TDOS divide up a disk into 4 sections. Section 1 takes up 1K of your disk or data pack for the "BOOT BLOCK" which contains the information required to load the operating system.

The next 12K is where the operating system is stored, if the diskette format allocates space for SYSTEM TRACKS. These tracks are loaded into memory and executed to boot the operating system.

Section 3 is the directory. This is where program names and their location on the disk or data pack are stored. The size of the directory varies, depending upon the format selected.

The remainder of the disk or data pack is available for program storage.

CARE AND HANDLING OF DISKS

Not all disks are compatible or interchangeable between systems. When you purchase blank disks, make sure they are for the type of drive you are using. If you have a single sided 160K disk drive, you can use any single sided, double density disks. If you have a double sided 320K or 360K disk drive, you can use any of the available double sided, double density disks. Either type must, of course, be of the 5.25" variety. Owners of a Quad Density 720K disk drive should use 3.5" double sided IBM type disks. DO NOT purchase the newer HIGH DENSITY or HARD SECTORED disks as these will NOT work on your ADAM disk drives. It is recommended that you not use single-sided disks in double-sided drives.

When you purchase software on a diskette to be used with TDOS, make sure the software is ADAM compatible. If it is not, you must convert it to ADAM format. The CDC-80 program is available (included MicroInnovations hard drive systems) to convert between Adam, IBM, Zenith, and TRS-80 CPM formats.

Remember that although disks are durable, they cannot withstand abuse. A coffee spill or the ravages of cigarette smoke and ash on any of the sensitive exposed areas, such as the head slot, is a disaster. Computer disks must be handled wisely. The following instructions below give tips for good disk care.

Always insert and remove disks carefully. Don't force them into the drive and always be careful when closing the drive latch.

Always store disks in their sleeves, in a clean place, away from electrical currents, dust, liquids and extreme heat or cold.

Before you turn off your computer or disk drive, remove all disks from your drives. Never try to remove a disk while the drive is running.

Always make and store away backup copies of disks separate from the originals. Use write protect tabs to protect your master back ups.

Never allow anything to touch the exposed surface of the disk. This includes fingers, dust, and food.

Keep disks away from magnets and magnetic fields. Many electrical appliances such as refrigerators, telephones, TV's or monitors, vacuum cleaners or even your ADAM memory console and/or printer have magnets.

Keep disks away from temperature extremes.

Never bend a disk; never use a sharp or pointed object on the surface of the disk. It's a good idea to write on your disks with a felt tip pen only. Never use a pen or pencil!

A sheet of write-protect tabs comes with every box of disks you may buy (other than the 720K disks which have a built-in tab) for the sole purpose of protecting your software. To write protect a disk, carefully remove the adhesive tab from the paper. Fold the tab (adhesive side to the disk) over the write protect notch so that it completely covers the notch. All four corners must be flat. Improper application of the write-protect tab can cause disk jamming in the drive.

If you remove the write-protect tab, you can once again write to the disk, alter it's files, or reformat it.

ERROR MESSAGES

TDOS issues two types of error messages. The first occurs when TDOS has detected has detected a physical disk error. The messages may be issued when you have entered a command at the AO> prompt or during the execution of a transient program such as Wordstar. The second kind of error only occurs at the command prompt and generally involves some sort of a command syntax error.

When TDOS detects a physical disk error, it displays the following message:

```
TDOS Error on A: error description
File: filename.typ
Retry, Abort, Fail?
```

The first line tells you the drive where the error occurred and the type of error. The second line tells you the name of the file associated with the problem (the problem may be internal to the file or it may be associated with the file's directory entry). Under some circumstances, there may be no filename reported. Finally, the third line describes what actions may be done to correct the error. Those are:

Retry - Try one more time to read or write to the disk in error. For Coleco disk and tape drives, you must open the drive and adjust the media for the retry to work.

Abort - This option aborts whatever program is running and returns to the command prompt. Use this option when all else fails.

Fail - This option terminates the bdos function in progress and returns to the calling program with an error flag set. Properly written programs will test this flag and take the appropriate action. Poorly written programs will not and should be aborted.

Press "R", "A", or "F" as desired. For some types of errors, only the Abort, Fail options are available. The most common errors are:

Read Only - Your program attempted to write to a disk that had been set to read-only status. Abort or Fail.

File Read Only - Your program tried to write to a file that had been set to read-only status. Abort or Fail.

Invalid Drive - Your program tried to select a drive that is not connected to your system. Abort or Fail.

Unknown Error - An unknown physical drive error occurred on a Coleco disk or tape drive. Adjust the media before retrying. Hard disk errors can also cause this message. Retry, Abort or Fail.

CRC Error - A Coleco disk or tape drive generated a CRC error. Adjust the media before retrying. Retry, Abort or Fail.

Missing Block - TDOS identified a missing sector on a Coleco disk or tape drive. Adjust the media before retrying. Retry, Abort or Fail.

Missing Media - TDOS could not find a disk or tape in your Coleco drive. Retry, Abort or Fail.

Write Protected - Your program attempted to write to a Coleco disk drive that had a write-protect tab placed on it. Remove the write-protect tab before retrying. Retry, Abort, or Fail.

Drive Error - A problem occurred in the disk drive mechanics such as using a hard sectored disk or the heads jammed. Adjust the media before retrying. Retry, Abort or Fail.

SASI Error nn - An attempt to use a MicroInnovations SASI device (floppy or hard disk) generated an error, code nn. Consult ERRCODES.DOC for an explanation of the error. Retry, Abort or Fail.

The remaining errors are generated at the command prompt. No corrective action can be taken with these errors. You must reissue the command that generated the error. These errors include:

No Room - You attempted to run a program that is larger than available memory. Do not retry.

File Exists -	You attempted to rename a file to a name used by another file. Use a different name.
I/O Error -	The directory or the disk itself to which you are writing is full or there was a disk error which you decided to Fail. Put in a new disk and try again.
fn.tp Not Found -	The file selected for a copy, type or rename was not found. Select a new disk or correct the filename. TDOS tells you the name of the file that it could not find.
Bad Syntax -	You violated the syntax of one of the built-in commands. Examples include using wildcards with type or rename or neglecting to have the source different from the destination in a file copy. Check your documentation and retry the command.

INSTALLATION

Before installing your TDOS system, your first task should be to make a backup copy of the distribution disk or data pack. It is suggested that a program such as FILEMANAGER or QUICKOPY be used to create a block copy of the full disk or data pack to a newly formatted disk or data pack.

If you don't have one of these, (or any other EOS program capable of doing a BLOCK copy) you can still create a backup of this disk or data pack.

Once you have finished, store your original TDOS data pack or disk in it's sleeve or plastic case and place it in a safe place.

To install your TDOS system, insert the distribution disk into your disk/tape drive and pull the reset switch. If you are using an 80 column monitor, be sure to turn on your TV set as the default distribution version is in 40-column mode.

Type DIR<cr> and select the appropriate installation file based on your system:

40 columns	40TDOSxx.COM
80 columns	80TDOSxx.COM

Note that the 'xx' will be 2 numbers indicating the revision number of the system.

After selecting the installation file, simply type its name and press RETURN:

40tdos41<cr>

NOTE THAT YOU DO NOT TYPE THE '.COM'

The installation program will start by scanning your system to determine the presence of disk/tape drives, memory expansion, hard drive, clock calendar, etc. Be sure all your drives are turned on.

You will first be asked which drive you want to install your system on. At this point, remove the distribution disk and insert a formatted disk into the drive you normally boot from.

You will then be asked about user modifiable options like disk drive sizes, logical positioning of ram card, I/O byte changes, keyboard translations, Smart key menus, etc. If in doubt, do not change anything. Except for disk drive sizes, all other changes are optional.

If you are running serial or parallel devices (like a printer or modem, you may be required to change to I/O byte (see that subject) or make modifications to the port settings.

Check the documentation of your serial card for the appropriate setting for serial port 1. 44H is the most common setting and is, for example, used by the Eve SP-1. If your documentation does not tell you what ports the serial card uses and 44H does not work, try the alternative port settings until the peripheral attached to the serial card give an indication that the card is working. Trying to access the serial port through a wrong address will harm neither the computer or the serial card. Do not use port 5CH if you have an AdamLink modem. The AdamLink modem also uses those ports and so your serial card should not use that port in order to avoid conflicts.

Repeat this procedure for serial port 2 if required. Once again check the documentation of your serial card or peripheral for the appropriate settings. The settings for Serial port 2 (base port 54H, 19200 baud, 8 bits per word, 1 stop bit and no parity) are already setup to work with the Orphanware 80 column display adapter. If you have the Eve version, change the base port to 4CH and the baud rate to 9600. 8 bits per word, 1 stop bit and no parity are fine for the Eve 80 column unit.

If you are unhappy with your initial settings, you can run the installation program again and set them differently.

If everything goes normally, you will get a TDOS installed message. At this point, pull the reset switch and your TDOS system is ready for use.