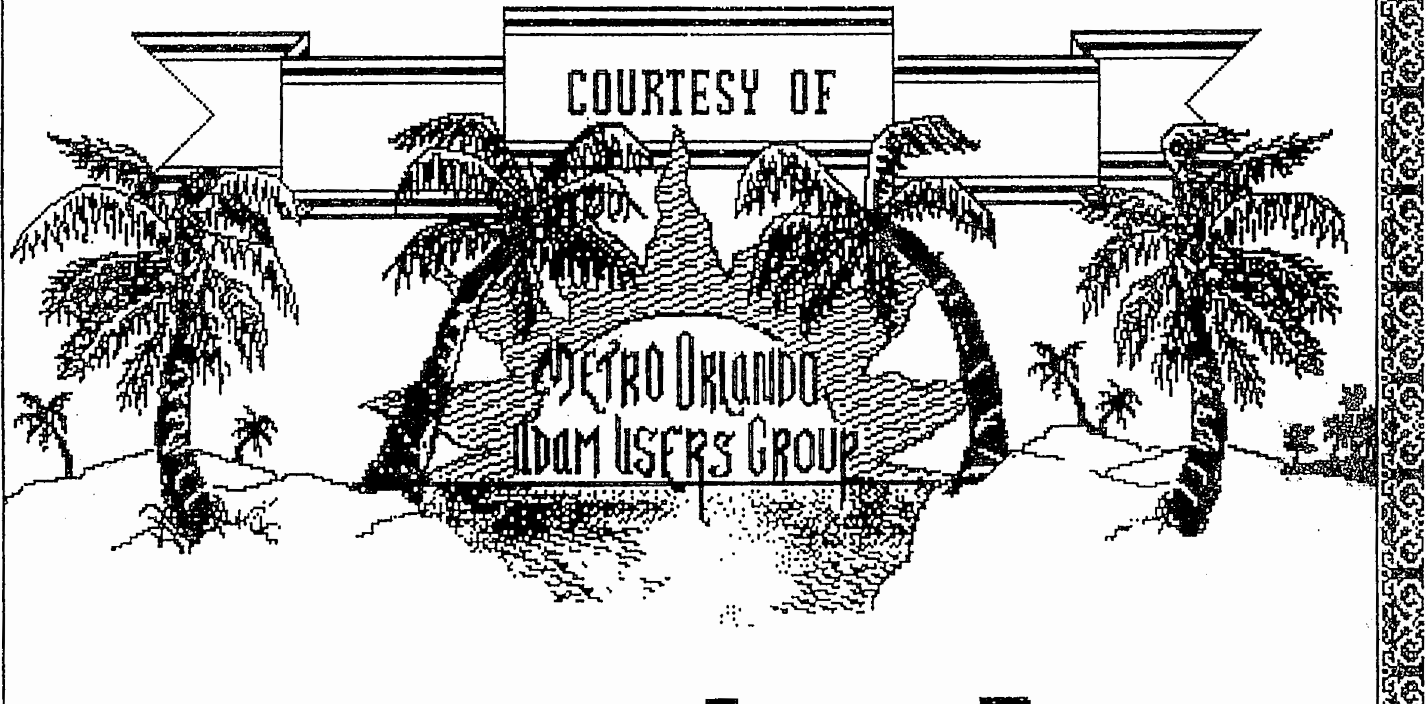
ADAMCON 03 SOUVENIR EDITION

TDOS 4.58 MANUAL

SOUTH BEND, INDIANA - AUGUST 1991



PRODUCED by JAMES POULIN

Cover Design: PJ Herrington

COPY NUMBER 007

This document package being offered in hopes that the recipient will become interested in and proficient with the advanced CP/M operating system T-DOS. These documents may be reproduced and shared with your friends and fellow AUG members. MOAUG (Metro Orlando Adam User's Group) extends the following offer to the recipient of this document package; a copy of Rob Friedman's TDOS.GIV, EXTRA.GIV and MODEM.GIV disk packages. The TDOS.GIV disk is a self-booting version of T-DOS that is set up to run on the basic plain vanilla ADAM system. After booting T-DOS, the user can run the appropriate install program (for 40 column or 80 column displays) and configure T-DOS to operate on his/her individual ADAM system. The documentation contained herein is complete and is all you will need to learn to use the T-DOS operating system. For your copy of the aforementioned disk sets, send inquirery to the address below and **include the Copy Number found on the cover sheet of the folder** and include media type desired (disk or digital data pack).

Write to: James Poulin
1146 Manatee Drive
Rockledge, FL 32955

AMERICAN SOFTWARE COMPANY

SPECIALIZING IN SOFTWARE & HARDWARE
FOR THE ADAM COMPUTER
P.O. Box 11097
Albany, NY 12211-0097

Telephone: (518) 459-4114
24 Hours A Day - 7 Days A Week

DIDN: 930001 **Printout Date: 26 November 1993**

This document package was originally produced by JAMES POULIN of the Metro Orlando Adam User's Group (MOAUG) for distribution at ADAMCON 03, held on August 1st through August 4th, 1991 in South Bend, Indiana as a door prize and souvenir.

The **AMERICAN ADAM USERS GROUP has written permission from JAMES POULIN** to reproduce this **T-DOS Manual**, and has done so under the auspices and courtesy of the **AMERICAN SOFTWARE COMPANY**, both of which are **owned and operated by Herman H. Cherney.**

The **AMERICAN ADAM USERS GROUP**, through the **AMERICAN SOFTWARE COMPANY** extends the following offer to the recipient of this document package; a copy of Rob Friedman's **TDOS.GIV, EXTRA.GIV** and **MODEM.GIV** disk packages.

The **TDOS.GIV** disk is a self booting version of **T-DOS** that is set up to run on the basic plain vanilla **ADAM** system.

After booting **T-DOS**, the user can run the appropriate install program (for 40 column or 80 column displays) and configure **T-DOS** to operate on his/her individual **ADAM** system.

The documentation contained herein is **COMPLETE** and is **ALL YOU** will **need to learn** to use the **T-DOS Operating System.**

For your copy of the aforementioned disk set, call or write to:

AMERICAN SOFTWARE COMPANY
P.O. Box 11097
ALBANY, NY 12211-0097

TELEPHONE: (518) 459-4114

and include the **DIDN Code Number located on the same line as the PRINTOUT DATE appearing on this page.**

Please note that DISK SETS ONLY are provided under this offer!

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.

TABLE OF CONTENTS

1-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. Your 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.

2-FEATURES

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.

It 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.

3-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 the 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.

CP/M 2.2 and now TDOS are machine language routines that overwrite the EOS with a brand new OS and then stay there until you clear them out by pulling the reset, or power down.

The CCP, BDOS, and BIOS are the 3 components of the operating system. See those help subjects for more information.

The operating system 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.

The operating system resides on disk and is loaded into memory when you boot your system. The CCP and BDOS are also copied to an unused area in the Video Ram. 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 utilities. 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.

4-CONSOLE COMMAND PROCESSOR (CCP)

The CCP is the low level interface between the operating system and the user. It has rudimentary features which help you manage your files. See RESIDENT COMMANDS for more information.

In addition, the CCP is responsible for processing your batch files (see that subject) and loading in user programs.

5-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 warmboot the system each time you change disks.

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.

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 FDH	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 add	none
10	Read Console Buffer	DE=buffer add	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	none
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
38	Get DMA Address		HL= DMA address
39	Get permanent media		HL=Permanent media bit-map
40	Write Random zero fill	DE=FCB Address	A=Directory code

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

ADVANCED TDOS FUNCTIONS

FUNC	NAME	PARAMETERS	RETURN CODE
41	Get Time	DE=user buffer	(HL)=time
42	Set Time	DE=user buffer	none
43	Get File Stamp	none	(HL)=stamp
44	Use File Stamp	none	
45	Get Program Code		
46	Get Drive Label (name)		A=Directory code
47	Set Drive Label (name)	DE=FCB (Name)	A=Directory code
48	Change file's user #	DE=FCB	Dest. user # in S1

49	Direct Bios Call		E=Vector number
			Status returned
			HL=Parameters From Bios

6-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.

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.

Acessing 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, 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 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
5	0F	List Output	C=char
6	12	Punch Output	C=Char
7	15	Reader In	none
8	18	Home Disk	none
9	1D	Select Disk	C=disk
10	21	Select Track	BC=Track
11	24	Select Sector	BC=Sector
12	27	Select DMA	BC=DMA
13	2A	Read Disk	none
14	2D	Write Disk	none
15	30	List Status	none
16	33	Sector Translate	DE=Vector BC=Track
	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 Keys	none
33	63	Key Display On	none
34	66	Key Display Off	none
35	69	Auxiliary Out	C=Character
36	6C	Auxiliary In	none
37	6F	Aux Out Status	none
38	72	Aux In Status	none
39	75	Bios Time	Time in (HL)

40	78	Read one Block	A=device
			DE=Buffer BC=Block Number

41	7B	Write One Block	A=device
			DE=Buffer BC=Block Number

42	7E	AUX2 Out	C=char
43	81	AUX2 In	none
44	84	AUX2 Out Status	none
45	87	AUX2 In Status	none
46	8A	Flush Console	none

7-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 BIOS handles 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	Expansion Ram
B	Disk Drive 1
C	Disk Drive 2
D	Tape Drive 1

In order to LOG into disk drive 1, simply type: B:<ret>. Then the system prompt changes to:

```
B0>
```

The 0 indicates that user 0 has been selected; this is the default. There are 16 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 1000K or 7000K 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. There is no more need for the CP/M USER command. 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.

See the CD command for more on users.

8-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.

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 'O' 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*	MY?????G.D*C	(no character
B6:M*.COM	B6:M*G.COM	allowed after *)
M*.?O?	M*.:O?	(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.

9-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>.

The next section 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.

10-SUBMIT OR BATCH PROCESSING

TDOS has the ability to perform several operations from an instruction file called a SUBMIT file. There is a BATCH.SUB file on this disk; have a look at it by typing:

```
A0>TYPE BATCH.SUB
DIR
1:
DIR 0:*.COM
A0>
```

The file consists of commands that you would normally type at the keyboard.

To run the BATCH file, simply type its name (without the SUB). Then sit back and watch it perform all the specified instructions: First it gives you a directory of all files on drive A0>. The it logs in to user 1 and shows a directory of all the .COM files in user 0. Then the batch file terminates. Remember to type 0:<ret> to get back to user 0.

You can also pass parameters to a SUBMIT file by including a '%' followed by a number from 1 to 9 in the file itself. Lets 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 A0 while working on GAMES.Z80 in B6.

The cold boot sequence will also look for a PROFILE.SUB on drive A0. This can be handy to execute any task you want when you turn your system on.

11-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                      ADAM.COM
COPY.COM                      CPMADAM.COM
SYSGEN.COM                    CONFIG.COM
SUBMIT.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.

12-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.

You can CREATE, REMOVE, or UPDATE the directory names

CD.COM should be placed in drive A0 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> /m
```

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.

In order to use the CD command to move between user areas, log back into drive A0 and type:

```
A0>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.

13-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.

14-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.

15-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.

16-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:

'current drive user 0'
'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 A0
```

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 B0 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 A0. 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 B0, and so on with drives C and D.

If you do not require CURRENT DRIVE USER 0 on the path, remove it via the path command. This will speed up access to your programs in A0:. Note that this command can also be put in your PROFILE.SUB file.

17-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 ^DELETE marks the end 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.

18-HARD DRIVES

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).

For serious programmers, this may be a very rewarding investment.

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.

19-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   IOBYTE

Address 3 contains the I/O byte. It is referenced by the BIOS on every input or output function in order to access the appropriate device.

04   DEFAULT DRIVE/USER

Address 4 contains the current user area in the high nibble and the current drive in the lower nibble.

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.

20-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>
```

21-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.

22-TYPE

The type command will type a file to the screen 24 lines at a time, pausing at each screen. Simply press RETURN to carry on. If you want to abort the display, press RETURN twice in a succession.

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 please 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
```

23-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 the line count.

24-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. It's 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:

```
C0> COPY *.COM B:
```

Copy all the .COM files from drive C0 (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 A0:B*.COM B2:
```

Copy all .COM files starting with 'B' from A0 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.

25-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 *.*".

26-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:]ufn1 ufn2
```

Where ufn1 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.

27-CLEAR SCREEN

On occasion, you may wish to clear the screen. Simply type

```
CLS
```

and it is done instantly.

28-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).

29-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: A0>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.

30-COMMAND BUFFER

All the commands that you type at the CCP prompt are stored in a buffer. In addition, any program that uses the READ CONSOLE BUFFER function will also have a record of 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 more previous commands. To scroll forward through the list of commands, press the down arrow key.

Recalled commands can be edited and re-executed. (see line editing, below).

31-LINE EDITING

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 unless you are prepared to lose the editing facility (see keyboard translations).

For compatibility, the following keys can be used instead of the function keys:

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

32-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 only ones supported by TDOS; explanations of their use will be given as required.

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)

33-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.

34-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      DnnnnnnnntttE12R0123456789ABCDEFR123
```

```
          ||       |  || ||               ||
          ||       |  || ||               ||>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 '.'.

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. Thes words are treated as file names and are parsed accordingly.

35-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 13K 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 included to convert between Adam, IBM, Zenith, and TRS-80 CPM formats.

Remember that although disks are extremely 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.

37-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.

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

The TDOS prompt: A0> 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 0.

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

If you make an error, TDOS will give you a meaningful message such as FILE NOT FOUND or SYNTAX ERROR. This is a great improvement over CP/M's approach of repeating the command with a question mark.

38-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.

CU.DOC

CU.COM is a program to move files from one user number to another. You can specify one file or a group of files using wildcards. Files can be located on any drive or user number but CU.COM will not move a file from one drive to another. CU.COM will prompt you before each change to make sure that you really want to move the file.

```
SYNTAX:       CU [do:]FILENAME.TYPE [n:] [/N]

Where    d is the drive designator (optional),
         o is the source user number, and
         n is the destination user number.
        /N means don't ask before each change
```

Filename.typ may be ambiguous, same as with the ERA command. You can omit either the source or the destination user number. However, the source and the destination user number must be different.

EXAMPLE:

```
                CU 0:*.ASM 3: /N
```

This would transfer all files of type ASM from user 0 to user 3 without asking before each change.

R/O and SYS files can be moved. R/O files will be R/W after the move.

THEORY OF OPERATION:

The first byte of each file specification in the directory is the user designation. CU.COM uses a special BDOS call that is unique to TDOS version 4.5 to change that user designation. Unlike other Change User programs, CU.COM will work correctly when used in a conjunction with a debugger such as Z8E or DDT.

CU.COM uses TDOS BDOS function 30H which has the following syntax:

```
     LD          DE,FCB          ;FCB points to a file control
                                 ;block containing the name of the
                                 ;file to change. Byte 13 of the
                                 ;FCB must contain the destination
                                 ;user number. You must already
                                 ;be logged into the source user
                                 ;number.
     LD          C,30H
     CALL        BDOS
```

DSKSZ.DOC

Disk Size v 2.0

By Tony Morehen / AJM Software

This program temporarily changes the disk parameters that the TDOS operating system uses to determine the size of the various drives that you have attached to your Adam computer. It should only be used with TDOS version 4.0 or greater.

What this program does is let you use different size disks with double-sided or 80 track disk drives. With the Micro-Innovations floppy, it even allows you to change the order in which the drive reads the data off the disk. This allows you to read 720K disks formated by the E&T or Orphanware 720s, disks that are otherwise incompatible. Why the need of this you say? Well, since I have a OBS <now CL Digital> 720 drive and a 320 drive, I still need a way to read all my OLDER CP/M disks. If I try reading an older 160K disk in my 320K drive.. it doesnt see the disk correctly. Hence i get disk errors..<like the drive trying to read the back side of a single sided disk> You can see the problem. What JAWS2 does is let you pick the size of your disk so you can read smaller drives or set up your drive to be a smaller size.

The command sequence is like this:

```
    Micro-Innovations Disk Size V3.0
         copyright AJM software


Patch CP/M Drive
(A to P,Z = quit)? <--- enter the letter of the drive to be changed, ie B


MAXIMUM SIZE OF DRIVE B

    0 - NO CHANGE

    1 - 145K (std Coleco
              single-sided)  <-- this is what you would use to
                                  use older standard 160K disks
                                  This also uses 1K blocks <more
                                  on that in a bit.

    2 - 254K (medium-sized
              double-sided)  <---this is the old EVE double sided
                                  disk drive. It also uses 1K
                                  blocks.
    3 - 304K (full-sized
              double-sided)  <--- this is the standard 320K drive
                                  from OBS and CL Digital systems
                                  <note: OBS also made a 360K
                                  drive, but this version doesnt
                                  have any mention of that drive.
                                  If you own a 360 OBS drive,
                                  contact CL DIGITAL SYSTEMS.>

    4 - 702K (quad density)  <--- this is the standard 720K drive
                                  that OBS and CL Digital systems
                                  convert up to.

    5 - 714K (quad density,
              2 system tracks) <-- this disk doesnt use the
                                  Standard Patched Adam System.
                                  Contact CL Digital or Tony
                                  Morehan for further info.

Enter Choice :1

Patch CP/M Drive
(A to P, Z = quit)? z
```

What this did was patch a drive to a standard 160K disk. This will now let you read, write, or format as if you had only a SS drive. The reason to switch the larger disks are that certain programs copy directly from one drive to another. Since they copy files from block to block, it is needed on some programs to have the same size disk block minimum. In the 320, and 720, the block size is 2K. On the 160 and the 256K drive..the size is 1K. If you use certain block swap programs <Like TRSIBM.COM> you would need to have 1K blocks. If you need a drive set up to 1K, you could use a 720 but tell it is only a 256K drive. I've done it and it works.

Remember when you are finished using disks of the temporary size, you MUST use DSKSZ20 to reset your drive to its normal configuation or else you will get errors reading your standard disks.

FF.DOC

FF.COM

This program will search all valid drives and user areas for a specified file. For example, enter:

```
     AO>FF NAME*.*<ret>
```

to locate any file commencing with 'NAME' on valid disk drives and any user area. It doesn't matter what user area or drive you are in when you start, it commences at AO: and searches all valid disks and user areas. If it finds a matching filename, FF.COM will display the filename, starting with the drive and user area. When its search is complete, FF.COM will display a count of the matching filenames.

You must supply a filename. If you do not FF.COM will display a short help screen.

You can get the entire directory of all drives and all available user areas by saying:

```
     AO>FF *.*<ret>
```

FF.COM will send the results of its search to the printer if you include the /P option:

```
    AO>FF *.COM /P<ret>
```

FF.COM will normally ignore system files unless the /S option is included:

```
     AO>FF *.* /S<ret>    will display all files including system files
```

The /S and /P options may be combined in any order.

Files with attributes set will be flagged as follows:

```
              -s-- = system file
              -sro = system file also set for read-only
              --ro = read-only file
```

If you specify a drive as part of the filename, FF.COM will search only that drive for the requested file. The specified drive can be any drive connected to the system, not just those included in the valid drive vector (see below). Use this option to search floppy drives for a lost file.

As noted above, FF.COM only searches valid drives. As distributed, those drives are A through D, inclusive. If you want to change the default drive search, edit the VALDRV vector in the TERMINAL.Z80 file. Set the bytes associated with the drives you want to search to 0 and the bytes for other drives to OFFH. Only hard disk volumes and ram disks should be included in the valid drive vector;

including floppy drives will result in errors unless a disk is in the drive when FF.COM is used. For instructions in the use of TERMINAL.Z80, see TDOSINST.DOC.

FORMAT.DOC

FORMAT34

Introduction

FORMAT34.COM prepares a diskette or tape for use with the TDOS operating system. It initializes the directory and writes a boot track to the disk so that TDOS will automatically be loaded after a hard reset. FORMAT34.COM can optionally analyze floppy disks or tapes for defective tracks. This version of FORMAT is a comlete replacement for the standard FORMAT.COM and SYSGEN.COM distributed with CP/M 2.2 for the Coleco ADAM. FORMAT34.COM works equally well with single- or double-sided drives or with Coleco or Micro Innovations' floppy disk drives.

FORMAT34.COM must be used with any new diskette or tape and with any diskette or tape that may have been used under the EOS operating system (ie. used in conjunction with SmartWriter, SmartBasic or other non-TDOS program). Note that the use of FORMAT34.COM will destroy any information already on the diskette or tape.

Program Requirements

TDOS 4.5 or greater

Backing Up FORMAT34.COM

It is strongly recommended that you create a working copy of FORMAT34.COM before attempting to use the program. The distribution diskette should be used only to create additional working copies.

FORMAT34.COM is not copy-protected so making a backup is a relatively simple procedure. Use PIP.COM or COPY.COM to copy FORMAT34.COM from the distribution disk to another TDOS formatted disk or tape that has at least 2K of free space available. STAT.COM can be used to verify that 2K is available on the destination disk.

Examples:

```
PIP C:FORMAT34.COM=A:FORMAT34.COM<return>

or

COPY A:FORMAT34.COM C:<return>
```

The two examples copy FORMAT34.COM from drive A to drive C.

Using FORMAT34.COM

When used for the first time, FORMAT34.COM will ask you about the sizes of the floppy disk drives that you have. Note that FORMAT34 must run under TDOS Version 4.5 or greater.

FORMAT.DOC

The syntax for the FORMAT.COM program is as follows:

```
FORMAT34 d:{/S}{/Q}{/V}{/I}
```

where d: is the TDOS drive to be formated and {} are optional parameters.

The parameter options have the following effect.

/S - By default, FORMAT34.COM automatically formats and installs the system on the diskette. This option eliminates the format process. This option is used to sysgen a disk that has already been used. It will not destroy any data already on the disk.

/Q - Quick format diskettes. This option will quick format and sysgen diskettes that have already been formatted by initializing the directory and rewriting the system tracks.

/I - repeats the installation process that occurs when you use FORMAT34.COM for the first time. Use this option if you change the size of disk drives in your system or if you add more disk drives.

/V - By default, FORMAT34.COM will not analyse a diskette just formatted for defective tracks. This option forces the verification process.

Remarks:

1) If you do not specify the drive on the command line, FORMAT34.COM will ask you which drive to format.

2) FORMAT34.COM will not format hard drives but it will format RAM disks.

3) More than one option can be specified on the command line.

4) A control-C (^C) at any prompt will abort FORMAT34.COM and return you to TDOS.

5) After the format process is complete, you will be asked if you want to format another diskette or tape. "Y" or "y" will repeat the format process with all options unchanged. If you wish to change any options including the drive to format, exit and restart FORMAT34.COM with the desired options.

Examples:

FORMAT34 F: - FORMAT34 will format and initialize drive F:

FORMAT34 - you will be asked what diskette to format. FORMAT34.COM will then automatically format and, if installed, sysgen a blank diskette in the specified drive.

FORMAT F:/V - FORMAT34.COM will format, verify and initialize the blank diskette in drive F:.

FORMAT34 /S/I - Format32.COM will run through the installation process and then initialize/sysgen the diskette. You will be asked for the drive to format.

Installing FORMAT34.COM

As mentioned above, when run for the first time, FORMAT34.COM will ask several questions about the sizes of drives that you have connected to your Adam system. If you should ever add or change disk drives, repeat the installation process using the /I option.

The installation process is follows:

1) Run FORMAT34.COM for the first time or with the /I option.

2) Answer the questions about the size of each of the disk drives connected to your system. Note that each screen is similar in appearance. Check the drive designation before responding to each screen.

3) FORMAT34.COM will then ask you to insert a diskette on which it can save the installed version of itself. Note that the diskette should have at least 5K free. Also, you do not have to change disks, FORMAT34.COM can save itself back to the diskette from which it was originally run. FORMAT34.COM will ask you for the TDOS drive to which to save the revised FORMAT34.COM. Make sure that you specify the drive using its logical TDOS name (ie. A: to P:). FORMAT.COM will erase any copy of FORMAT34.COM that is already on the diskette to which it is saving itself. Press <return> when the appropriate diskette is inserted in the specified drive.

4) The instllation process is now complete. FORMAT34.COM will now ask you to insert a blank diskette for fomatting. Press 'Z', 'z' or control-C if you do not wish to format a diskette. Otherwise, put a blank diskette in the specified drive and press <return>.

5) FORMAT34.COM will now format (and optionally verify) and sysgen the blank diskette.

INITDIR.DOC

T-DOS can keep track of the last time a file was modified and accessed if your hard disk directory has been initialized with INITDIR.COM. Date and time stamping work best if you have a real time clock but regular use of the DATE program (see below) will also work well.

INITDIR.COM initializes directories by clearing the fourth entry of each directory sector and sets the first byte of that entry, its user area, to 21 hex. This entry contains the time and date stamps of the other three directory entries in the sector. Note that after a directory is initialized, the number of available directory entries are reduced by 25 percent. For most hard disk users this reduction is not a problem because you generally run out of space well before you run out of directory entries. However, if you intend to put a lot of small files on your hard disk, you should not run INITDIR on the partition containing those files.

The format of the date/time entry is as follows:

```
DB   21H                        ;date stamp identifier
DB   90H,04H,16H,22H,10H        ;file 1 modified date/time in BCD
DB   90H,04H,16H,22H,10H        ;file 1 last accessed date/time in BCD
DB   90H,04H,16H,22H,10H        ;file 2 modified date/time in BCD
DB   90H,04H,16H,22H,10H        ;file 2 last accessed date/time in BCD
DB   90H,04H,16H,22H,10H        ;file 3 modified date/time in BCD
DB   90H,04H,16H,22H,10H        ;file 3 last accessed date/time in BCD
```

If you wish to enable file time and date stamping, you must initialize the directory of each of your hard drive partitions (generally drives A: through D:), using the program INITDIR.COM. The syntax for that program is:

INITDIR D: where D is the letter of the drive to be initialized.

The drive letter is required. The directory can already have files in it. INITDIR will move the directory entries around to accomodate the time and date stamps. Also, INITDIR will not initialize a directory if there are not enough directory entries remaining.

From this point on TDOS will keep track of the creation date and update date of all files. If you have an Eve clock/calendar or equivalent and a 64K or larger memory expander, TDOS will read the clock for the current time. If not, it will read a data area in the BIOS for the date. The program DATE.COM allows you to put the correct date in that area (It will also set the clock/calendar if available). The DATE program is self explnatory; just respond to the prompts. If you do not have a clock/calendar, DATE should be put into your PROFILE.SUB in drive A0: so that it will be run whenever TDOS is started. It is also good practice to run DATE.COM several times during a session, so that files created later in the session will have a different time stamp from those created earlier.

MLOAD

Description:

MLOAD is a transient command used to convert a "HEX" file (the output of an assembler) into an executable "COM" file.

In its simplest form, MLOAD takes a program that is in valid Intel hexadecimal format machine code and converts it into a command file that can be executed. The file becomes filename.COM (the hexadecimal file is filename.HEX). This form may be used as a replacement to the Digital Research "LOAD" command.

The program may also be used to overlay portions of a "COM" file with patches assembled to a new "HEX" file.

Syntax:

```
mload [<outnam=>][<file1.type>,]<file2>[,<file3...>] [bias]
```

<outnam> is the optional output file name.

<file1.type> is an optional non-HEX file to be patched by subsequently named HEX files. Specifying the filetype enables this function.

<file2> specifies the file to be loaded, unless a file1.type has been specified. In that case, it specifies a HEX file to load.

<file3> specifies HEX file(s) to load.

<bias> is the offset within the saved image to apply when loading the file.

MLOAD with no arguments prints a small help message. This message is also printed whenever a command line syntax error occurs.

ZCPR-style du specifications are fully supported, for both input and output files. The following command lines are permissible:

```
b3>mload a4:myfile.com=0:bigfil,b6:patch1,c9:patch2
a6>mload b5:=c3:mdm717.com,mdmpatch
```

Only the optional drive and user spec and the primary filename may be specified. The output filetype is derived exclusively from the 3-byte string at 103h within MLOAD.

If no output filename is specified, it will be derived from the first input filename, with filetype of "COM", if not otherwise specified (this default filetype may be patched directly into MLOAD via DDT or with MLOAD itself, using a patch file - its location is at 103H in MLOAD.COM). Note that a command line of the form "C:=<FILENAME>" will place the output file on the "C" drive with the same primary filename as the input file.

This feature obviating any need to use DDT. The first (and only the first) filespec (after the "=", if used) may be non-HEX and the filetype must be specified. Examples:

1. "MLOAD MYPROG" loads MYPROG.HEX and writes the output to MYPROG.COM.

2. "MLOAD WS.COM,WSPATCH" loads WS.COM, overlays it with WSPATCH.HEX, and writes the output to WS.COM.

3. "MLOAD BYE5=BYE5.COM,B5KP-1" loads BYE5.COM, overlays it with B5KP-1.HEX, and writes the output file to BYE5.COM.

4. "MLOAD WS.OVR,ORRPATCH" loads W S.OVR and patches it with "OVRPATCH.HEX".

Filenames may contain drive/user specs, and must not contain wildcards. Input filenames must be separated by comas, and a space is required between the last filename and the optional bias.

A load information summary is printed at the successful conclusion of the load. Any errors in loading will generally include the name of the file in question.

This program is a replacement for the CP/M "LOAD" program. Why replace "LOAD"? well... LOAD.COM has a few deficiencies. For example, if the hex file's origin is above 100h, LOAD.COM prepends blank space to the output file to insure it will work as a CP/M transient. It doesn't care if the file is not intended as a CP/M transient and it also doesn't like hex records with mixed load addresses. For example, one that loads below a previous record; which is a perfectly legitimate happenstance. Also, LOAD.COM can load only one program at a time, and has no provision for a load bias in the command specification. Finally, there is no provision for user specification of the output file name.

In its simplest form, MLOAD's syntax is identical to LOAD.COM. There should be no problem in learning to use the new program. The only significant difference here is that, under LOAD.COM, all files are output starting at 100h, even if they originate elsewhere. MLOAD outputs starting at the hex file origin (actually, the first hex record specifies the output load address). The bias option may be used to override this.

An example should clarify this. Suppose you have a file that loads at 1000h. LOAD.COM would save an output file that begins at 100h and loads past 1000h (to wherever the program ends). MLOAD will save an output file starting from 1000h only. If, for some reason, you need the file to start at 100h in spite of its 1000h origin (I can think of several circumstances where this would be necessary), you'd have to specify a bias to MLOAD. Using this example, "MLOAD MYFILE OFOO" would do.

Note that this program re-initializes itself each time it is run. If your system supports a direct branch to the TPA (via a zero-length .COM file, or the ZCPR "GO" command), you may safely re-execute MLOAD.

Error Messages:

```
        Command line syntax error
                Error in command line, re-enter and try again.
        Ambiguous file name: <filename.ext> not allowed.
                Ambiguous files are not allowed.
        File <filename.ext> not found.
                Input file could not be found, check spelling and try
                again.
        Disk full.
                No room on disk for output file.
        Directory full.
                No room in directory for output file.
        Premature end-of-file in <filename.ext>
                Error in input file, correct error, and try again.
        Checksum error in <filename.ext>
                Error in input file, correct error, and try again.
        Can't close <filename.ext>
                Problem closing file, disk may be in Read/Only status.
        Memory full while loading <filename.ext>
                Error in input file, correct error, and try again.
        Format error in file <filename.ext>
                Error in input file, correct error, and try again.
        Writing <filename.ext>, nothing loaded
```

PATH.DOC

PATH - TDOS path manager

PATH displays and sets the TDOS path. The command syntax for PATH is:

```
PATH                             - displays the current path
PATH [DU1:] [ DU2:] ... [DU5:] - sets the path, maximum of 5 DU:
PATH ON                          - enables the current path (default)
PATH OFF                         - disables but does not change the current
                                   path
PATH ?                           - displays PATH's internal documentation
```

where du#: is a disk D and user U element to be included in the current path. D may range from 'A' through 'P' and U may take any value from 0 through 31. The $ sign may be used in either position to signify the default disk and/or user. The colon is optional.

SORTDIR.DOC

SORTDIR will sort all directory entries in alphabetical order and repack the directory, filling empty and deleted file entries with OE5H. Although this has a marginal effect on file access, DIRECTORY programs will behave much faster.

SYNTAX: sortdir d: [/n][/d]

The drive MUST be specified

/n removes NAMES created by CD.COM

/d removes date stamping
- you must re-enable date stamping with INITDIR.COM

use /n/d or /d/n for both.

IOBYTE.DOC

IOBYTE.COM is used to make a temporary changes to the I/O byte. If you want to make a permanent change, you must reinstall TDOS.

The I/O byte is a control byte in zero page. It consists of 3 2-bit codes and 2 1-bit codes which control the input/output functions of your system. These bits are set during TDOS installation to default values for the keyboard, screen, printer, and other external devices. IOBYTE.COM uses menus that are identical to the TDOS installation I/O byte menus.

IOBYTE.COM first reads and then displays the current setting of the I/O byte. Those settings are as follows:

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.

UNDELETE.DOC

UNDELETE.COM
A SMART FILE UNDELETE UTILITY
By Guy Cousineau

1990-01-28 1.2	Fixed for SECTOR PER TRACK > 255 Fixed wild card mask to find high-bit-set Fixed double show of files > 512K Moved CP/M 2.x check to program start Incorporated Disk reset routine Changed U=UNDELETE to T=TAG Speeded up duplicates check Added RAMTOP to protect from HUGE directories
1990-01-11 1.1	Fixed sector translate bug Fixed for disks with more than 16 SPT Added wild card mask to screen files
1989-08-02 1.0	First released to public domain

Special thanks to Bridger Mitchell for a selective file reset utility for logging in fixed drives. His full subroutine is included intact and is fully transportable to any other program. See the large chunk of code written entirely in lowercase.

Thanks also to Howard Goldstein who was of invaluable help in debugging the preliminary versions of this utility.

PREAMBLE

First a few explanations about deleted files and directory structure. CP/M maintains a 32-byte directory entries for files. The first 16 bytes are similar to a File Control Block (FCB):

```
FCB OFFSET     0 1 2 3 4 5 6 7 8 9 A B C D E F
CONTENTS       d n n n n n n n n t t t e s1s2rc
```

The drive code goes in position 0
bytes 1-8 contain the file name
bytes 9-11 contain the file type
byte 12   contains the extent
bytes 13-14 are internal control bytes s1 and s2
byte 15   contains the number of records in this extent

So what's an extent? Consider it as the number of chapters in the files, and the record count as the number of pages in this chapter. The working of extents is a fairly complex exercise depending on your disk architecture. The difference between an FCB and a directory entry is the first byte. Once the directory is written to the disk, there is no need for the drive code so instead, the BDOS places the user number in byte 0.

When a file is deleted, the BDOS replaces that user number with an E5. This is the signal for a deleted file entry which can be re-used by another file. Unfortunately, the E5 also wipes out the old user number.

UNERA.COM was a welcome addition to the CP/M public domain and was very useful in recovering accidentally deleted files. It looked for files with the E5 code in byte 0 and replaced them with the default user number. It had, however, a few drawbacks in that it allowed unerasing files which would duplicate entries already in the current DU and unerased files which overlapped the same allocation blocks as other active files.

Hence the creation of UNDELETE.COM. This utility works in the following fashion:

1. find all deleted files (or mask provided on command line)
2. mask out files with no EXTENT 0 (incomplete)
3. mask out files with allocation overlaps
4. mask out files with the same name as active files (slow part)
5. prompt for tagging of files to recover
6. undelete the tagged files

SYNTAX: UNDELETE [d:][afn]

DETAILED EXPLANATIONS

1. If no drive is specified the current drive is used. The program LOGS the selected drive out and then back in to force a directory read. Each directory record is read and scanned for deleted files, building table entries in RAM for each found file. If a file name was specified on the command line, table entries are created only for deleted files matching the AFN supplied. If too many files are found, the program will abort. Thsi might only happen on a 64K directory with 1/2 its files deleted.

2. Print first report skipping all entries corresponding to further extents of a file. This fashion of reporting effectively masks out large files for which extent 0 has been overwritten but other extents still exist in the directory.

3. The table is re-scanned and each allocation block checked against a copy of the allocation vector. If there is overlap, the file flag is set to disable recovery. If the allocation is unused, it is set to used to prevent recovery of other files which may use the same allocation. A second report is then printed.

4. Check the current drive/user for matches of the deleted files remaining in the table. These files may be undeleted later if the corresponding active file is renamed. A third report is then printed showing only the files which are safe to undelete.

5. Print the files out one by one and ask the user to tag them with a 't' or a 'T'. After the files are tagged, the user is asked for confirmation: a 'u' proceeds and a CONTROL-C aborts. In order to abort the tagging process and jump to the CONFIRM stage, just press ^C.

At every stage of the analysis, a printout is given of the FILES FOUND SO FAR. This way, you can see those files you want to recover but might be unable to because of internal conflicts. If at any time, the number of recoverable files becomes 0, the program aborts. A ^C will also abort the program at the CONTINUE prompts.

USE CONSIDERATION

Although this utility is smarter than UNERA.COM it can fail if a file with more than one directory entry is missing its second entry. Under normal directory operations, this is an impossible situation since directory entries are filled in from the start of the directory. Thus if extent 1 is missing then extent 0 must be missing as well.

The ALLOCATION overlap can also fail if a sort-and-pack program was used to re-order the directory entries. Under normal operations, it is impossible for the second of 2 deleted files (using the same allocation blocks) to be the UNDELETEABLE one. With SAP, there is no way of accurately determining which file can be undeleted.

Both the above checks can be thrown out of whack if a file was manually deleted by a disk editor (like DUU or ZAP). A variety of other unusual happenings may affect program behaviour.....tread carefully.

If there are 2 copies of the same deleted file, they will show up as 2 entries in the TAG menu. To recover both and check which is the most current, start by tagging only one version and rename it to another UNUSED name. Run UNDELETE again and tag the second version. Check them both out and delete the one you don't want to keep.

How can this happen you say? Pull out your text editor (the one that creates .BAK files) and follow these steps: (My editor is VDE)

```
    VDE a.a
    this is a.a version 1                     [type this in]
                                              [exit]
    VDE a.a                                   [re-edit the file] this
    this is a.a version 2                     [change the 1 to 2]
                                              [exit]
```

Now I have just what i expect; 2 versions of the same file, but with different names. But I want to go back to version 1; so as i often do:

```
    ERA a.a
    REN a.bak a.a
```

After typing or doing whatever with a.a, I delete it as well. Later, I want to get it back and pull out UNDELETE which reports 2 files called a.a and both recoverable. If I tag them both, I can be in trouble. Although both are undeleted, the BDOS will only be able to find the FIRST one when i try to open it. If I try a rename or delete, it will find both and confuse me even more. So I should tag one, rename it, and run undelete again.

Although UNDELETE is smarter than UNERA, you should verify all recovered files before using. Particularly in the case of .COM files, it is wise to tread safely with back-up disks when testing the recovered file.

Since UNDELETE recovers files to the current user area, you should select the user number which is appropriate to the recovery attempt. Hey, you can even use UNDELETE to move a file from one user to the next by deleteing it from one area, logging into another user, and undeleteing it.

The utility has been tested on a CP/M 2.2 system and should behave correctly on any Z-80 CP/M 2.x. It has been tested with 1,2, and 4 K allocation blocks on media with 1 and 2 byte allocations. Although it has performed with no problems on these media, it is not guaranteed to work in non-standard environments. The utility will only run on a BDOS which reports a "2?" HEX when asked for version. CP/M 3.x does not like programmers writing directly to disk via BIOS calls.

Use the program in good health, please send bug reports to:

Guy Cousineau
1059 Hindley Street
OTTAWA Canada
K2B 5L9
OTTAWA RCPM (613) 952-2289

TDIR.DOC

TIME DIRECTORY PROGRAM

TDIR.COM, the time directory program for TDOS 4.5 and later displays the date and time that a file was created or last modified as well as the time and date that the file was last accessed. Time and date stamping must be enabled using the INITDIR command before TDIR will display any dates or times.

TDIR.COM supports full wild card card specifications, Drive & User:

```
A2>TDIR <CR>
A2>TDIR B6:*.?Z? <CR>
A2>TDIR 0:HELP.* <CR>
A2>TDIR C: <CR>
```

Four options are available when preceded with a '/':

/a lists files matching the afn which have not been archived.

/$ matches files in all users regardless of the current user in effect.

/1 forces TDIR to display one file per line, showing both the last modified and the last accessed dates. This is the default display for TDIR1-nn.COM, where nn is the version number.

/2 forces TDIR to display 2 files per line, showing only the last modified date for each file. This is the default display for TDIR2-nn.COM.

Options can be combined in any order. For example, "/$/a" and "/a/$" are equally valid.

Examples:

```
A2>TDIR c: /a            all unarchived files on drive c
A2>TDIR c: /$            all users on drive c
A2>TDIR c: /$/a          all unarchived on c
A2>TDIR *.com /$/a       all unarchived COM files
```

TDIR can also search and display a directory based on creation or update dates. This option must be the last in the list:

```
A2>TDIR >90-04-12        files created after April 12 1990
A2>TDIR <90-04-12        files created before April 12 1990
A2>TDIR >>90-04-12       files accessed after April 12 1990
A2>TDIR <<90-04-12       files accessed before April 12 1990
```

Using >> or << will use the second date in the listing. Note also that the < option means less than or equal to. All options above can be used to create a variety of reports:

```
A2>TDIR b6:*.com /a >90-04-12      all COM files on B6 which have not been
                                   archived and created after April 12 1990

A2>TDIR c: /$ >>90-04-12          all files in all users areas on drive C which
                                   were accessed after April 12
```