DATEL UTILITIES

FENTON INDUSTRIAL ESTATE,
DEWSBURY ROAD, fenton,
STOKE-ON-TRENT TEL 0782 273815
FAX. (0782) 264510

smart cart™

An 8K or 32K
pseudo ROM cartridge

instruction manual

DATEL UTILITIES

SMARTCART & UTILITIES.

This manual contains instructions for the Smartcart 32K plus the Ramdisk and Diskmate. Some of the instructions will only be relevant if you have that particular utility.

Smartcart was conceived designed and produced by Datel Electronics. C 1987.

All programming for Ramdisk and Diskmate are by Dosoft. C 1987.

DATEL UTILITIES

SMART CART - INSTRUCTIONS FOR USE

Smartcart is a lithium battery backed 32k Ram cartridge for the 64/128 enabling the user to create custom plug in cartridges without the need for an eprom programmer. Low power CMOS memory is used and a battery life of about 5 years is estimated.

The smartcart is best described as "Pseudo Rom". By this we mean that it is can be programmed just as Ram but then can be made to act like a Rom cartridge. It is a great aid in program development. Also, existing utility 8-ROM cartridges may be transferred into SMARTCART and modified to suit individual requirements.

Four banks of 8k are available, which can hold up to 4 separate 8-ROMS at one time. Advanced programmers can design systems for large applications using all four banks at one time.

INSTALLATION

The cartridge fits into the expansion port of your CBM 64 or 128. Looking from the front of the computer, this is at the rear right. NEVER insert or remove the cartridge with the computer switched on, or damage will likely result. With the power OFF, insert the cartridge into the expansion port, with the button to the right (looking from the front).

When the computer is switched on, the system is configured as if an 8-ROM cartridge was in place - 8k is removed from basic and the powerup message will show 30719 BYTES FREE. In addition, the I/O #2 slot is open at $DF00 and Smartcarts CONTROL REGISTER is onboard at $DE00 (56832).

THE CONTROL REGISTER

This new, write only register is the means by which Smartcart is controlled. Bits 0 and 1 control which of the four banks are enabled. Bit 2 switches all the banks off, leaving computer Ram in the processors address space. Bit 3 controls whether the enabled bank acts as Ram (read or write) or Rom (read only) memory. The remaining bits are unused.

BIT	DECIMAL VALUE	
0-1	0,1,2 or 3	Bank enable bits
2	4	EXROM - switches all banks out
3	8	Write enable

Examples:

```
POKE 56832,0 : REM enable smartcart bank 0 as ROM
POKE 56832,8 : REM enable smartcart bank 0 as RAM
POKE 56832,1 : REM enable smartcart bank 1 as ROM
POKE 56832,9 : REM 8+1 : enable smartcart bank 1 as RAM
POKE 56832,4 : REM switch out all banks
```

IMPORTANT

The control register is WRITE ONLY. If you PEEK the register a random value will be returned, and will also be written to the register. This may have unexpected results, so remember, do not PEEK the register or display it on a machine code monitor line. The best way to keep track of the cartridge configuration is to keep its value in a variable which can be altered and poked in as required. When powering down the computer, ensure that the cartridge is write protected. To be quite certain of this, hold your finger on the reset button as you switch off the computer power. This completely avoids any possibility of data corruption during the transition from computer to battery power.

AUTOSTART CARTRIDGES

When the CBM 64 is switched on, program execution commences at $FCE2 (64738). The stack pointer is reset and decimal mode is cleared. The system then checks for the presence of an autostart cartridge by checking loctions $8004-$8008 (32772-32776) against the 8-ROM mask (cbm80). If found, program control is passed to the cartridge via the indirect address at $8000-$80001 (32768-32769). This is the cartridge COLDSTART VECTOR. The cartridge WARMSTART VECTOR passes control to the cartridge when you press RUN/RESTORE.

Typical cartridge autostart routine:

```
8000 09 08              COLDSTART VECTOR
8002 25 80              WARMSTART VECTOR
8004 C3 C2 CD 38 30     8-rom MASK (cbm80)
8009 LDA #$00
     STA $D016          SWITCH ON VIC
     JSR FDA3           INITIALISE CIA CHIPS
     JSR $FD50          CLEAR LOW MEMORY AND SET TOP OF BASIC MEMORY
     JSR $FD15          INITIALISE KERNAL RAM VECTORS
     JSR $FF5B          INITIALISE VIC CHIP AND SCREEN EDITOR
     CLI                CLEAR INTERRUPTS
     JSR $E453          INITIALISE BASIC RAM VECTORS
     JSR $E3BF          INITIALISE BASIC
     JSR $E422          PRINT POWERUP MESSAGE
     LDX #$FB
     TXS                SET STACK POINTER FOR BASIC
     JMP $E386          BASIC WARM START
```

This is a standard initialisation routine which sets up the system as normal, except that the top of basic is lowered to account for the presence of the cartridge. Programmers normally add extra code which sets up the ram vectors ($0300-$0332) for basic extensions, fastloaders etc, before the basic warmstart jump.

If you are new to this kind of programming, the best way to learn is by studying the code of existing cartridges to see how the system is altered to suit the needs of specific applications.

THE OPEN I/O SLOT

This is a 256 byte page at $DF00-$DFFF which is not normally used by the CBM64. The last page of any Smartcart bank ($9F00-$9FFF) is also decoded at $DF00-$DFFF and remains in place even when all banks are switched out. System vectors can be redirected to this area to allow bank switching of smartcart as required, to produce a "floating" cartridge system which is transparent to normal operation, being switched in and out when required. Our DISKMATE, which is available separately to operate in smartcart, uses this system to give full access to all of the computers memory.

The I/O slot and control register follow the same rules as the rest of the I/O area as regards banking ie setting both loram and hiram of location 1 to zero will switch them out of the processors address space. The smartcart banks themselves follow the rules of the basic Rom - if basic is switched out then so are the smartcart banks.

SMARTCART - BANK LOAD/SAVE ROUTINE

Type in and save the following basic listing. It will load or save any Smartcart bank. The program is set up to work with disk device 8, but can be used with tape by changing line 100 to read DEV=1.

The program can also be used to save the contents of a standard 8-ROM utility cartridge for later installation into Smartcart. In this case select any bank number for the save. There is no need to type in the REM statements.

```
10 IF A<>0 THEN 2000:REM EXIT FROM LOAD
20 SYS 64789: REM RESET KERNAL VECTORS
30 SYS 58451: REM RESET BASIC VECTORS
50 POKE55,0:POKE56,128:CLR:REM LOWER TOP OF MEMORY
100 DEV=8:REM FOR TAPE CHANGE TO DEV=1
110 REG=56832:REM CONTROL REGISTER
120 A=0
200 PRINTCHR$(147)"SMART CART BANK LOAD/SAVE ROUTINE"
220 PRINTCHR$(17)CHR$(17)"1. LOAD A BANK"
240 PRINT:PRINT"2. SAVE A BANK"
300 PRINT:INPUT"SELECT 1 OR 2";A
320 IF A <>1 AND A<>2 THEN120
400 PRINT:INPUT"WHICH BANK (0-3)";B
420 IF B>3 OR B<0 THEN 400
500 PRINT:INPUT"FILENAME";F$: L=LEN(F$)
520 IF L=0 AND DEV<>1 THEN 500
530 POKE REG,B OR 8 :REM SWITCH IN SELECTED BANK AND UNPROTECT
600 IF A=1 THEN LOAD F$,DEV,1
625 REM AFTER LOAD,EXECUTION CONTINUES FROM FIRST PROGRAM LINE
1000 REM SAVE ROUTINE
1010 SYS(57812)F$,DEV,1:REM SETUP SAVE PARAMETERS
1040 POKE172,0:POKE173,128
1050 POKE780,172:POKE781,00:POKE782,160:REM SETUP ADDRESS POINTERS
1060 SYS62941:REM KERNAL SAVE ROUTINE
2000 POKE REG,B AND 3:REM WRITE PROTECT BANK
2010 IF DEV=1THEN END
2020 OPEN15,DEV,15
2030 INPUT#15,E,E$,E1,E2
2040 CLOSE15
2050 IFE=0 THEN END
2060 PRINTE;E$;E1;E2:END
```

Machine code programmers can save and load banks ($8000 - $A000) via a monitor. Remember that the control register ($DE00) should not be read and must be written indirectly via a subroutine, not by display and modification of a memory line.

NOTE: certain replacement kernal systems and disk turbos switch out the system roms (including smartcart) before saving or loading data. Such systems should be switched out before saving or loading banks.

Most standard 8k utility cartridges eg Mikro Assembler, Stack Help etc., will operate in Smartcart without modification. Simons basic is a 16k cartridge which uses a different type of bank switching and cannot be used in Smartcart. Likewise Epyx Fastload and Quickdisk. Games cartridges take complete hold of the system and cannot be saved by normal means.

SMARTCART - BANK CONTROLLER

Type in and save the following basic listing. This is a special bank controlling iinitialisation routine which allows you to enable any smartcart bank on pressing the reset button in conjunction with a keypress.

```
100 Z=0:POKE56832,8: REM ENABLE BANK 0,WRITE PROTECT OFF.
120 FORI=0TO149:READ A:POKE32768+I,A:Z=Z+A:NEXT
130 POKE56832,4: REM SWITCH BANK OFF
140 IF Z<>19295 THEN PRINT "ERROR IN DATA!"
500 DATA 9,128,94,254,195,194,205
501 DATA 56,48,162,255,120,154,216
502 DATA 169,0,141,22,208,32,163
503 DATA 253,32,98,128,32,91,255
504 DATA 32,159,255,32,228,255,162
505 DATA 4,160,253,201,0,240,9
506 DATA 41,3,240,32,160,225,234
507 DATA 234,170,169,252,72,152,72
508 DATA 120,160,6,185,71,128,153
509 DATA 34,0,136,16,247,76,34
510 DATA 0,142,0,222,136,208,253
511 DATA 96,169,128,141,132,2,32
512 DATA 83,228,32,191,227,32,34
513 DATA 228,162,251,154,76,134,227
514 DATA 160,32,185,47,253,153,19
515 DATA 3,136,208,247,152,153,2
516 DATA 0,200,208,250,160,85,153
517 DATA 0,2,200,192,167,208,240
518 DATA 169,60,133,178,169,3,133
519 DATA 179,169,160,141,132,2,169
520 DATA 8,141,130,2,169,4,141
521 DATA 136,2,96,140
```

When you run the program, an autostart sequence is loaded into bank 0, which coldstarts a bank according to which key is depressed on pressing the reset button. Hold down key 0, 1, 2, or 3 and press the reset button to enable bank 0, 1, 2, or 3 respectively. If no key is pressed the system will coldstart with no bank enabled. Keyys Ø, A, B, and C have the same function as the number keys.

This routine may be used by programmers as part of the initialisation of a custom 8-ROM cartridge.

Once this routine is installed, 8-ROM cartridges installed in banks 1,2,and 3 may be enabled as required. DISKMATE (avaiable separately) occupies bank 0 and has a similar coldstart system built in, therefore this and three other 8-ROMS may all be instantly available.

NOTE:

If you program Smartcart with a faulty 8-ROM start, you may find that the system locks up unrecoverably. If this occurs the cartridge may be drained by removing the case and lifting the blue tab near the battery, which disconnects the battery from te Ram chip.

Be aware that an 8-ROM (cbm80) mask may be present in computer ram at $8004-$8007. This can cause a system crash when RUN/RESTORE is pressed. Poke 32773,0 to disable.

SMARTCART 32K - RAMDISK

This program converts your 32k Smartcart into a Ramdisk. Program and sequential files can be saved and loaded from or to Smartram with the same commands used for accessing a real disk drive. Because Smartcart is battery backed, all data remains intact when you power down at the end of a session, ready for instant access at a later time.

INSTALLATION.

Ensure that the computer is switched off. Insert Smartcart into the expansion port and switch the computer on. From disk: LOAD "RAMDISK",8 and RUN. From tape: press SHIFT/RUNSTOP in the normal way. The ramdisk will load and automatically install itself into Smartcart. The normal CBM powerup screen will be displayed together with the ramdisk DIRECTORY:

```
DATEL SMARTCART SC01

100 BLOCKS FREE
```

THE DIRECTORY:

This consists of the disk NAME (Datel Smartcart) and version number, a list of the files contained in the ramdisk, and the number of BLOCKS FREE, which indicates the amount of free space left for data storage. One block contains 254 bytes of storage space. An empty disk contains 107 free blocks, giving 107*254 = 27178 bytes or characters of storage space. This space is entirely contained within Smartcart and does not use any of the computer ram. Each FILE ENTRY consists of 1) the number of blocks used by the file 2) the name of the file and 3) the filetype (PRG - program file used for LOAD and SAVE operations. SEQ - sequential file used for Basic strings etc in conjunction with the OPEN,PRINT#,INPUT#,GET# and close commands (see later).

Whenever you want to see the directory, just type DIR (return), or $ (return). You can obtain a hardcopy of the directory on you printer with:

```
OPEN 4,4:CMD4  (return)
DIR            (return)
PRINT#4:CLOSE4 (return)
```

The ramdisk can hold a maximum of 16 directory entries.

LOADING AND SAVING PROGRAMS.

Ramdisk acts as a peripheral device just like a disk drive or Datassette. Each peripheral has its own DEVICE NUMBER to distinguish it from other devices. For example, a real disk drive is normally accessed as device 8, and tape has a device number of 1. The Ramdisk device number is 7, but this can be changed if you wish. Programs are loaded and saved using the normal LOAD and SAVE commands. Each program must have a unique filename, and you must specify the correct device number which tells the computer that the Ramdisk is to be accessed:

LOAD "PROGNAME",7 will load a basic program from the ramdisk into computer memory.
SAVE "PROGNAME",7 will save the basic program in memory to the ramdisk.
VERIFY "PROGNAME",7 will compare the program in memory with the program in the ramdisk.

Possible errors:
FILE NOT FOUND: You tried to load a program which was not in the ramdisk. Note that you must type the filename exactly.
FILE EXISTS: You tried to save a program with the same filename as one which already exists in the ramdisk.
DISK FULL: There was not enough space left in the ramdisk to hold the program which you tried to save.
DIRECTORY ERROR: The directory already holds the maximum of 16 entries and cannot hold any more.
LOAD ERROR: Indicates that the program in the ramdisk is corrupted and cannot be loaded correctly.
VERIFY ERROR: Indicates that the program in memory is not the same as the program in the ramdisk.

SECONDARY ADDRESS FOR "LOAD".

Basic programs do not require a secondary address. However, machine code programs saved via a machine code monitor must be loaded back to the same area of memory from which they were saved:
LOAD "PROGNAME",7,1 ensures that this occurs. ",1" (the secondary address) directs the computer to load the program back into the same area of memory from which it was saved.

CHANGING THE DEVICE NUMBER.

The ramdisk device number is normally 7. You can change this if you wish by the following command:

```
DEV (new number)
```

eg: DEV8 changes the ramdisk device to 8, and all acccess to the ramdisk will require this device number. If you just type DEV without a number, the current ramdisk device number will be displayed. The ramdisk may be given any device number except for 0,2,and 3.

SENDING COMMANDS TO THE RAMDISK.

A number of special commands may be sent to the ramdisk to perform certain tasks which help to keep the ramdisk tidy, and allow files to be renamed and deleted if required. There are two ways of sending commands. 1) From within a program via the COMMAND CHANNEL (see later). 2) Directly via the "@" or "*" commands. The following commands are available:

@ display ramdisk error status.
I Initialise.
V Validate.
N: Newdisk.
S: Scratch a file.
R: Rename a file.

ERROR STATUS: Type @ (return) and the message 00,OK,00,00 will be displayed. If any error occurs during access to the ramdisk, the screen border will flash briefly, and reading the error status will return a descripton of the error which occurred.

INITIALISE: @I or *I. This clears any open channels within the ramdisk (see later).

VALIDATE: @V or *V. This tidys up the disk and frees any block which have been allocated to files which no longer exist in the directory. It also removes any files which have not been correctly saved eg if the disk became full during a save. Use this command whenever a file in the directory has an asterisk (*) against it, or whenever the total number of blocks do not add up to 107. This ensures maximum availability of ramdisk blocks.

NEWDISK: @N:DISKNAME. This completely clears the ramdisk of all files, and retitles the disk with the given name. NOTE: after entering this command all directory entries will be cleared and you cannot recover any files. Be certain that you no longer require any data before you new the disk. If you just want to change the disk name without clearing the directory, use the following command:

NAME "DISKNAME" (return).

SCRATCH: @S:FILENAME. This removes the named file from the directory and frees its allocated blocks in the ramdisk. NOTE: you cannot recover a file once it has been scratched, so be certain that you no longer require the file before scratching it.

RENAME: @R:NEWNAME=OLDNAME. This changes the name of a file in the directory, eg if a file is called "FRED" and you want to change it to "JOHN", enter @R:JOHN=FRED (return).
@ and * are interchangeable.

These commands may also be used with a real disk drive, by adding the device number when accessing the command channel. All further access will be to that device.
eg: @8 reads the error status of device 8. All further commands will be to that device. To access the ramdisk, enter @7 (or current ramdisk device number). Further commands will be to the Ramdisk.
@,$,S:,N:,V,I, and R commands can also be sent within a program by opening a file to the command channel eg:

```
10 OPEN15,7,15
20 PRINT#15,"I"
30 CLOSE15          See later for a fuller description.
```

One additional command, OLD (return) will recover a Basic program which has just been "NEWed" or after the cartridge has been reset.
To disable the Ramdisk enter SYS 64738

SAVE WITH REPLACE.

The Scratch and Save commands can be combined to allow you to overwrite an existing Ramdisk program as you save a new one. Syntax is as follows:

```
SAVE "@:PROGNAME",7
```

This will save a program named "PROGNAME" and will scratch any program with that name which is already on the Ramdisk.

WILDCARDS AND PATTERN MATCHING.

It can be tedious to have to type in the whole filename when loading or scratching programs. Two special characters can be inserted into a filename to make this process easier:

* means "ignore all further characters" ? means "ignore the character in this position"

Some examples will help to clarify this. Say you have a program on the Ramdisk which is called "LONG FILENAME". Instead of typing the name in full you could enter:

```
LOAD "LON*",7
```

This will load the first file in the directory with the first three characters "LON". All further characters will be ignored when the file is being searched for.

LOAD"*",7 will load the first file in the directory.

LOAD"???" will load the first file which has three characters in the filename.

@S:?? will scratch all files with two characters in the filename

@S:* will scratch EVERY file in the directory

Pattern matching cannot be used with the Save command. Be careful when scratching files because you can inadvertently scratch files which you want to keep.

TAPE TURBO.

Supplied with the tape version of Ramdisk only. This is a stand alone program which can be used independently of the Ramdisk. To enable the Tape turbo:

```
LOAD "TAPE TURBO",7 and RUN
```

Saves and loads to tape are increased in speed by about six times. Reading and writing of sequential files are approximately three times faster.

IMPORTANT. All tape turbos save to tape in a special format. This means that you cannot load programs which were saved at standard speed, or with other tape turbos, unless you first disable the tape turbo.

To disable the tape turbo enter SYS 52482 (return)
To re-enable the tape turbo enter SYS 52480 (return)

POWERING DOWN.

At the end of a programming session, the computer may be switched off and all data within the ramdisk will be retained under battery power ready for the next session. However you should note that if the power is switched off while the ramdisk is operating eg during a load, there is a small chance of data corruption due to electrical noise during the transition from computer to battery power. To avoid this completely, we recommend that you hold down the reset button on the cartridge as you switch off the computer power. This will completely protect the Smartram against corruption.

The supplied SMARTCART BACKUP program allows you to save and reload the entire contents of the Ramdisk from or to tape or disk. Tape users should load and run the Tape Turbo program before using this utility.

FILES AND CHANNELS

In addition to the normal LOAD and SAVE commands, Ramdisk supports the CBM sequential filing system, using the OPEN, CLOSE, PRINT#, INPUT# and GET# . This system is normally used to save variables and strings for later recall. An OPEN command to the Ramdisk takes the following syntax:

```
OPEN (file number),(device),(channel),"filename,filetype,mode".
```

FILE NUMBER. This can be any number between 1 and 255.
DEVICE. This is the Ramdisk device number, normally 7
CHANNEL. This can be any number between 2 and 14. Channel 15 is reserved for sending special commands (see later).
FILENAME. The name of the file you wish to open.
FILETYPE. Ramdisk supports three filetypes - PROGRAM files (P), SEQUENTIAL files (S), and USER files (U). User and sequential files are in fact identical in structure, differing only in their filetype designation in the directory. Program files differ in that they are recognised by the LOAD command. The first two bytes of the file are taken as the load memory address and the rest of the file is loaded starting at that address.
MODE. There are three modes - READ (R), WRITE (W) and APPEND (A). Read is used when you want to fetch data from an existing file. Write is used when you create a new file for sending data to. Append is used when you want to add data to the end of an existing file which has previously been closed.

```
eg: OPEN 1,7,3,"TESTFILE,S,W"
```

This opens a sequential write file in the ramdisk with the name "TESTFILE". If you miss out the filetype and mode, the ramdisk assumes that you want to open a sequential file for reading:

i.e: OPEN 1,7,3,"TESTFILE" is the same as OPEN 1,7,3,"TESTFILE,S,R"

The PRINT# command is used to send data to the file and takes the syntax PRINT#(file number),(data)

eg: PRINT#1,"TESTSTRING" , where 1 is the filenumber of the file to be written to.

INPUT#(filenumber) is used to fetch data from a file which has been opened for reading. It reads characters into a variable string until it reaches a carriage return (CHR$(13)) or a comma.
GET#(filenumber) fetches a single character. INPUT# and GET# can only be used from within a program. INPUT# can read a maximum of 88 characters. If it does not find a carriage return or comma before reading 88 characters the error "STRING TOO LONG" will be generated. It is important when writing strings to a file to ensure that strings longer than 88 characters are not used, otherwise you will not be able to read them back with INPUT#. Any file can be read with GET#, however, but this is much slower.

```
eg: 10 OPEN 1,7,3,"TESTFILE,S,R"
    20 INPUT#1,A$
    30 PRINT A$
```

File data which is not arranged in strings must be read character by character using GET# instead of INPUT#.
There is an oddity in Basic whereby GET# cannot read a CHR$(0) into a string. This causes problems when converting to ASCII. To get round this use: GET#1,A$: A=ASC(A$+CHR$(0)).
PRINT# automatically sends a carriage return along with the string. To suppress this the statement must be followed by a semicolon: PRINT#1,A$; will send a string without a carriage return.

CLOSE(filenumber) is used when you have finished with a file. This frees the filenumber and channel for later use. You must always CLOSE a file when you have finished writing to it, otherwise you may not be able to fully read the data from it at a later time.

The CBM 64 allows up to 10 files to be open at one time, to various devices eg printer, disk drive, modem etc. Up to four files can be open to the Ramdisk at one time. Each must have a unique channel number.

THE COMMAND CHANNEL

Also known as the error channel. This channel uses the reserved channel number of 15, and is used for two purposes - to send commands to the Ramdisk, and to read the ERROR STATUS of the Ramdisk.

SENDING COMMANDS. Commands are the same as described earlier for the @ or * direct mode command. Syntax: PRINT#(filenumber) "(command)"

```
eg: 10 OPEN2,7,15: PRINT#15,"S:TESTFILE"
```

This scratches the file called "TESTFILE". Filetype and mode are not necessary when opening the command channel.

READING THE ERROR STATUS. The command channel can be read as well as written to. Reading the channel returns a message describing the status of the drive after an input or output operation. If any error occurred this will be described.

```
eg: 100 INPUT#15,E,E$,E1,E2
    120 PRINT E;E$;E1;E2
```

result: 0 OK 0 0

This message consists of 1) The ERROR NUMBER (0 if all is well), 2) The ERROR MESSAGE (OK if all is well) 3) Track and sector numbers (for compatibility with Commodore DOS)

Common errors are: FILE NOT FOUND, FILE EXISTS, NO CHANNEL (you tried to write to a read file, or tried to access a channel which was not previously opened). The error channel should be checked frequently within a program to check that all is well.

A second means of checking for errors is by means of the STATUS WORD. This is a reserved Basic variable, ST. This will be zero if all is well. If an error occurs, it will have a value of 3. If you read past the end of a file, it will have the value 64. See the attached example programs for further information.Note that the value in ST is set after each access to a device.When transferring data between two devices, status should be preserved in a variable if this is to be checked after acccessing the second device.

SEQUENTIAL FILES ON TAPE.

Tape is a more limited medium than disk. Only two channels are available, Channel 0 for reading and channel 1 for writing. Seqential files only are available. Append cannot be used. Filetype and mode do not apply to sequential files on tape.

The following basic listing is intended to illustrate how sequential files operate, and some of the techniques used to transfer files from device to device, and shows how the error channel and status variable can be used to keep a check on how the input/output operations are progressing. Dont type the listing in, rather refer to it if you encounter any problems when writing your own sequential file programs.

```
10 REM TRANSFER A SEQUENTIAL FILE FROM DISK TO RAMDISK
15 OPEN4,8,15:OPEN3,7,15:REM OPEN ERROR CHANNELS OF DISK AND RAMDISK
20 OPEN1,8,2,"PROGNAME,S,R":GOSUB4000:REM OPEN READ FILE (DISK)
30 OPEN2,7,3,"PROGNAME,S,W":GOSUB5000:REM OPEN WRITE FILE (RAMDISK)
40 GET#1,A$:IFA$=""THENA$=CHR$(0):REM FETCH A CHARACTER FROM DISK
50 S=ST:IF(SAND3)THENGOSUB7000:GOTO5500
60 PRINT#2,A$;:IF ST<>0 THENGOSUB7500:GOTO5500
70 IF(SAND64)<>0THEN6000:REM CHECK FOR END OF FILE
80 GOTO40
100 REM
110 REM WRITE BASIC STRINGS TO RAMDISK
120 REM AND READ THEM BACK
122 DIMA$(100)
124 FORI=1TO100:A$(I)="TESTSTRING"+STR$(I):NEXT:REM CREATE STRINGS
140 OPEN3,7,15,"I":REM OPEN RAMDISK ERROR CHANNEL AND INITIALISE
150 OPEN1,7,6,"STRINGS,S,W":GOSUB5000
160 FORI=1TO100:PRINT#1,A$(I):IF ST THEN GOSUB7500:GOTO5500
170 NEXT:CLOSE1
180 REM
190 REM READ STRINGS BACK
200 OPEN1,7,1,"STRINGS,S,R"
210 INPUT#1,A$:PRINTA$:IF ST=0 THEN210
220 GOSUB7500:GOTO5500
300 REM
310 REM TRANSFER SEQUENTIAL FILE FROM RAMDISK TO TAPE
320 OPEN3,7,15
330 OPEN1,7,3,"STRINGS,S,R":GOSUB5000
340 OPEN2,1,1,"TAPEFILE":REM OPEN WRITE FILE TO TAPE
350 INPUT#1,A$:S=ST:PRINT#2,A$:IFS=0THEN350
360 GOSUB7500:GOTO5500
400 REM
410 REM TRANSFER A SEQUENTIAL FILE FROM TAPE TO RAMDISK
420 OPEN3,7,15
430 OPEN2,7,3,"FILE,S,W":GOSUB5000
440 OPEN1,1,0,"TAPEFILE":REM OPEN READ FILE TO TAPE
450 INPUT#1,A$:S=ST:PRINTA$:PRINT#2,A$
460 IF ST THEN GOSUB7500:GOTO5500
470 IFS=0THEN450:REM CHECK FOR END OF FILE
480 GOSUB7500:GOTO5500
4000 GOSUB7000:IFE<>0THEN5500:REM CHECK DISK ERROR CHANNEL
4020 RETURN
5000 GOSUB7500:IFE<>0THEN5500:REM CHECK RAMDISK ERROR CHANNEL
5020 RETURN
5500 PRINTE;E$;E1;E2:REM PRINT ERROR STATUS
6000 CLOSE1:CLOSE2:CLOSE3:CLOSE4:END:CLOSE FILES AND END
7000 INPUT#4,E,E$,E1,E2:RETURN:REM READ DISK ERROR CHANNEL
7500 INPUT#3,E,E$,E1,E2:RETURN:REM READ RAMDISK ERROR CHANNEL
```

DISKMATE - SMARTCART V2

This special version of our famous DISKMATE is designed epecially to work in SMARTCART and provides all the features of the original utility without exception. Two versions of the program are provided:

1. DISKMATE 1. For 8K and 32K Smartcart. Occupies Bank 0.
2. DISKMATE 2. For 32K Smartcart. Occupies banks 0 and 1.

Diskmates 1 and 2 are identical except that 2 includes fast file copy and disk backup. Instructions are as per the enclosed Diskmate manual with two additions:

1. Additional disk instruction in the monitor:

```
.@*(source)(destination)
```

Allows monitor Dissassemble, Assemble, Memory display and Transfer instructions to work with disk drive memory.

```
eg .@*88  diverts read and write to drive 8
   .@*89  read from drive 8, write to drive 9
   .@*08  read from computer memory, write to drive 8
   .@*8   read from drive 8, write to computer memory.
   .@*    restore default - read and write computer ram.
```

Use .@*88 to assemble or modify drive memory. Note that the compare command does not work with drive memory.

2. Bank control during reset. Similar to the bank controller as described in Smartcart instructions. Hold down the following keys while pressing the reset button to give the described effect.

```
   No key.     Initialise Diskmate as normal.
   Back arrow. Disable Diskmate and Smartcart completely.
   1,2 or 3    Coldstart with respective bank 1,2 or 3 enabled.
or A,B or C
    0 ,or @     Default system coldstart.
```

When Diskmate 2 is installed, bank 1 is set up to return to bank 0 on coldstart. Be aware that the default system coldstart, and warm start (run/restore), will be directed by vectors at $8000 and $8002 respectively if an 8-ROM mask (cbm80) is present in computer ram at $8004. This can cause the system to hang. POKE 32773,0 will disable an 8-ROM mask.

INSTALLATION OF DISKMATE INTO SMARTCART.

Diskmates 1 and 2 are supplied as Load/Run files.

```
LOAD "DISKMATE 1",8 and RUN
LOAD "DISKMATE 2",8 and RUN
```

Diskmate will automatically install itself into the cartridge. Because Smartcart is battery backed and effectively acts as Rom, the system will remain installed even when you switch off the computer or remove the cartridge. If during experimentation Diskmate's coding becomes corrupted, it will be necessary to drain the cartridge of data. To do this, take off the cartridge casing and lift the plastic tab near the battery. Diskmate can then be reloaded from disk.

SMARTCART BACKUP

Supplied with Diskmate and Ramdisk. This routine is an extension of the Bank Loader program listed in your Smartcart instructions, which allows you to transfer any or all Smartcart banks to disk or tape, and from disk or tape back into Smartcart.
The program is largely self explanatory, and is written in basic so the user can expand or modify the utility to suit his or her requirements.

When you save the whole of Smartcart, four separate 8K files are saved, which correspond to the four banks, 0,1,2,and3. Each filename is prefixed by the number of the bank from which it was saved. Example: a cartridge with the given filename CARTRIDGE will be saved in four parts as follows:

```
      0CARTRIDGE     1CARTRIDGE     2CARTRIDGE     3CARTRIDGE
```

When you reload a previously saved cartridge, note that you should enter the name only. Do not add the numerical prefix. This will be done automatically by the program. Remember that any data already in a bank will be destroyed when you load into that bank, so be sure to have saved it first if it is to be required later.

Individual banks may also be saved or loaded as required, in which case no number will be added to the filename. You must also specify which bank (0,1,2,3) is to be loaded or saved. The "X" option will bypass the bank switching register to allow saving of an existing 8-ROM cartridge which can be plugged in instead of Smartcart before loading Smartcart Backup. Many 8-ROM cartriges can be installed into Smartcart eg Mikro Assembler, Stack Help, Arrow. Others such as Epyx Fastload, Quickdisk,Simons Basic and game cartridges use a different bankswitching system and are not compatible.

TAPE TURBO.

Supplied with Ramdisk tape version only.

This is a stand alone tape turbo utility which can also operate independently of the Ramdisk. The program is supplied as a Ramdisk file and is present in the directory when you install the Ramdisk. To enable the tape turbo, enter:

LOAD "TAPE TURBO",7 (return) and RUN

The tape turbo sits high in the computers memory and is compatible with all basic programs and sequential files. Loading speed is increased by about 6 times for program files and about 3 times for sequential files.

IMPORTANT. All tape turbos save data to tape in a special format. Turbo data can only be loaded when the tape turbo is in memory and initialised. If you want to load programs recorded at standard speed it will be necessary to disable the tape turbo.

To disable the tape turbo enter SYS 52482 (return)
To re-enable the tape turbo enter SYS 52480 (return)

If you press RUN/RESTORE or press the cartridge reset button the turbo wil be disabled. Enter the above SYS number to re-enable the turbo.

Load and initialise the tape turbo before using Smartcart Backup.

DISKMATE - INSTRUCTIONS FOR USE

Diskmate is a cartridge based disk turbo system which greatly enhances the performance of your 1541, 1570 or 1571 disk drive. It uses no computer memory, and rarely, if ever, will it interfere with the normal operation of your computer. Diskmate also provides some useful toolkit commands, and a powerful machine language monitor.

SINGLE STROKE COMMANDS. A number of easy to use single stroke commands for load, save, directory and error channel are provided:

```
  /PROGNAME is equivalent to LOAD"PROGNAME",8
  ↑PROGNAME is equivalent to LOAD"PROGNAME",8 followed by RUN
  %PROGNAME is equivalent to LOAD"PROGNAME",8,1
  £PROGNAME is equivalent to SAVE"PROGNAME",8
  $, $8, or $9  will display the directory without corrupting memory.
  @, @8 or @9   will read and display the disk error channel.
  @ COMMAND STRING is the equivalent of OPEN15,8,15:PRINT#15,"COMMAND
STRING".
```

Example command strings @I, @S:NAME, @V, @R:NEWNAME = OLDNAME. See you disk drive manual for details of disk commands.

If you are using a two drive system, then device 9 can be accessed with the single stroke commands if you first read the error channel with @9. Subsequent commands will be to that drive.

FAST FORMAT. Diskmate will format a disk at very high speed.
Example format command: @N:DISKNAME,01
Upon entering this command, the message "ARE YOU SURE?" will be displayed. Press "Y" to continue or any other key to abort. If you are using a two drive system, remove the disk from the other drive before formatting the required disk, otherwise you may accidentally format the wrong disk (!), depending on which drive was previously accessed. Once in a while, the format will not "take" and the directory may appear odd. If this occurs repeat the process and everything will be okay.

DISK SAVE ENHANCEMENTS. The turbo save used in Diskmate is a little slower than some routines available, but this one will report any error which occurs, and is more widely compatible with different drives. The reported "@0:" bug in the CBM DOS has been bypassed, therefore SAVE"@0:PROGNAME" may be used. The previous program is scratched prior to saving the new program.

DISKMATE will calculate the number of disk blocks required by a program to be saved, and compare this with the number of blocks free on the disk. If there will not be enough room, you will be informed of this (if in direct mode). You will then have the option to abort, or continue saving after inserting another disk.

FUNCTION KEYS. The function keys have been defined as follows:

F1 is equivalent to LOAD"0:*",8,1 (ie load the first program on the disk)
F2 is equivalent to LOAD"0:*",8,1 followed by RUN
F3 display directory.
F4 change background colour.
F5 RUN

F6 change border colour.
F7 LIST
F8 enter machine language monitor (see later).

NOTE: the even numbered function keys (F2,F4,F6,F8) are accessed by holding the SHIFT key before pressing the appropriate function key.

LOADING FROM THE DIRECTORY. If you display the directory and then move the cursor onto a line containing a directory entry, you can load that program simply by pressing F1 (or F2 to run). This is a very convenient way of loading a program.

TOOLKIT COMMANDS

A number of extra basic commands are provided. To ensure compatibility with existing software, these commands only operate in direct mode. They will not work when a program is running. (the same applies to the function keys). Any of the commands may be abbreviated by typing just the first three characters of the command eg: MON will be interpreted the same as MONITOR.

OLD will recover a basic program which has been NEWed, or after pressing the reset button.

DELETE will delete a block of program lines. Syntax is the same as LIST except that the first line to be deleted must be specified.

```
example DEL 1000-2000 will delete lines 1000 to 2000 inclusive.
        DEL 1000- will delete from line 1000 to the end of the program.
```

LINESAVE will save to disk a section of a basic program.

example LIN"PROGNAME",8,1000-2000 will save program lines from 1000 to 2000 inclusive. Lines saved in this way may be loaded back as separate basic programs, or MERGED into other programs.

MERGE will load a basic program from disk and combine it with a program in memory. If two lines have the same number, the new line will replace the existing line. A program may also be merged with new line numbers.

eg: MERGE"PROGNAME",8 merges the program on disk with the prog in memory.

MERGE"PROGNAME",8,1000,10 will renumber the lines before they are merged, starting with line 1000 and incrementing in steps of 10. GOTO and GOSUB statements will not be renumbered.

Merging into a long program is a lengthy process, so be patient.

APPEND differs from MERGE in that the new program is tagged onto the end of the previous one. It also occurs at turbo speed. eg APP"PROGNAME",8 loads the program, starting at the end of the previous one. For append to be useful the program to be appended should have line numbers greater than the existing program.

AUTO provided automatic line numbering as you type in a program.

eg AUTO 1000,10 starts automatic line numbering at line 1000 and increments in steps of 10

To turn AUTO off, press return against a blank line. If you subsequently want to continue automatic numbering, type AUTO on its own, and numbering will continue from the last line number which was automatically displayed.

BOOT will load a machine code file and jump to the first address loaded. Eg if you have a program (say a basic extension) which you would normally load with LOAD"PROGNAME",8,1 followed by a SYS number, BOOT will perform the same function. Note that BOOT will only work when the program is started by SYSing to the first address loaded. Syntax BOOT"PROGNAME", or BOOT"PROGNAME",(device number).

BLOAD, BSAVE, BVERIFY are BLOCK commands. These are for machine code files, which may be loaded or saved to any location in memory.

examples:

BLOAD"PROGNAME" load from current device to the address from which the program was saved.
BLOAD"PROGNAME",8,C000 load starting at $C000 (hexadecimal notation).
BSAVE"PROGNAME",8,C000,D000 save memory from $C000 to $D000
BSAVE"PROGNAME",8,C000,D000,E000 save memory from $C000 to $D000 to load back at $E000

The BLOAD and save commands always save and load Ram locations.

DISKMATE may be disabled by typing OFF (return). Loads and saves will then be at standard rate. A SYS number will be displayed which will re-enable Diskmate. Very rarely, a program will not work with diskmate enabled. Often, typing SYS 64738 will allow the program to work without having to unplug the cartridge.

Many commercial programs will work with Diskmate. However you should note that some disks are loaded by special methods which will bypass the diskmate fastloader.

The new Diskmate Mk III cartridge now has file copy and disk backup facilities. See later for a description of these.

MACHINE LANGUAGE MONITOR.

DISKMATE has a powerful extended machine code monitor, designed for fast bidirectional examination and modification of any area of the computer's memory. To enter the monitor type MON (return) or press the F8 key.

BANK SWITCHING. On entry the monitor is in RAM mode. All the computer's Roms will be switched out when memory is accessed. To change modes enter "*". This will toggle and display the current mode, RAM or ROM.

EXAMINING MEMORY. Memory may be examined in Hex/Ascii, Disassembly, or Interpreted Ascii forms. Examples:

```
.D C000 D000  disassemble from $C000 to $D000
.M C000 D000  display memory from $C000 to $D000
.I C000 D000  interpret memory from $C000 to $D000
.D C000       disassemble one instruction at $C000
.M C000-      display memory from $C000 onwards
.D            disassemble from the current address onwards.
```

CONTROLLING THE DISPLAY. While memory is being displayed, ANY KEY will halt the display, from where any key will continue. F5 and F7 will reverse the direction of the display, therefore in either disassemby, memory or interpret modes you may scroll either up or down in memory. Press STOP to exit. Memory may be modified by typing over the byte to be altered, or by typing over the instruction if in disassembly.

OTHER MONITOR COMMANDS:

```
.A C000 LDA #$01
```

Assemble an instruction in standard 6502 mnemonics at $C000. The next memory address will be displayed ready for the next instruction.

```
.R
```

Register display. The program counter, A, X, and Y registers, and the stack pointer are displayed in Hex. The Status register is displayed in Binary. Registers may be altered by typing over the relevant, Byte or flag.

```
.F C000 D000 AA
```

Fill memory from $C000 to $D000 with the byte $AA

```
.H C000 D000 01 02 03
.H C000 D000 "STRING"
```

Hunt through memory for a sequence of bytes or an Ascii string. If any occurrence is detected, the address will be displayed.

```
.C C000 D000 E000
```

Compare the area of memory from $C000 to $D000 with memory starting at $E000. If there is any discrepancy, the memory locations will be displayed together with the bytes at those locations.

```
.T C000 D000 E000
```

Move memory starting at $C000 and ending at $D000 and move it to memory starting at $E000. This is an intelligent transfer - memory areas may overlap and may be moved either up or down in memory.

```
.G C000
```

Load registers as per the register display and start executing the machine code program starting at $C000. The program may end with either a BRK or an RTS instruction.

```
.N $C000
.N 49152                        .N %10101010
```

Number conversion. Displays a number in HEX, DECIMAL, and BINARY. If the number is a single byte value, the ascii character of that number will also be displayed.

```
.P
```

Direct output to a CBM printer. P is used as a prefix to another command eg:

```
.PM C000 D000
.PD C000 D000
.PH C000 D000 "STRING"
```

In each case, memory display will be directed to the printer (device 4). Hold STOP to quit. You will find this a very useful command.

```
.B
```

Can be very useful. Execute a Basic instruction from the monitor eg:

```
.BPRINT8*256
.BPRINT"TEST"
.PBLIST  (LIST to the printer)
```

```
.L"PROGNAME"
.L"PROGNAME",8,C000
.L"",1,C000   (relocated load from tape)
.S"PROGNAME",8,C000,D000
.S"PROGNAME",8,C000,D000,E000
.V"PROGNAME"
```

Load, Save, Verify. These commands are the same as the BLOAD, BVERIFY and BSAVE commands described earlier.

```
.X
```

Exit to basic.

The monitor uses a small area of memory from $0110 to $0140, and $0250 to $0258, which should not be overwritten.

FILE COPY. (Diskmate 2 only)

Enter COPY or @C to enter the file copier. This is very useful for transferring files from one disk to another. The file copier will handle up to 44 program, sequential or user files up to 247 blocks in length, but not relative, random or specially formatted files. Output may be to a second drive if required. If you select MULTI OUTPUT from the menu, several copies may be taken from each pass, if required. Your output disk or disks should be formatted prior to entering the file copier.

Diskmate's save routine uses a more efficient disk sector interleave, so you will often find that programs load faster after being file copied.

DISK BACKUP (diskmate 2 only)

Enter BACKUP or @B if you want to copy a whole disk. This is a very fast single drive whole disk copier. The process requires three passes and you will be prompted to insert your source and destination disk in turn. The destination disk need not be formatted in advance, but note that any data already on this disk will be lost. We recommend that you put a write protect tab on the source disk before starting to copy, to avoid losing data if you get the disks mixed up. This copier is not for commercial protected disks.

SMARTCART BACKUP

Supplied with Diskmate and Ramdisk. This routine is an extension of the Bank Loader program listed in your Smartcart instructions, which allows you to transfer any or all Smartcart banks to disk or tape, and from disk or tape back into Smartcart.
The program is largely self explanatory, and is written in basic so the user can expand or modify the utility to suit his or her requirements.

When you save the whole of Smartcart, four separate 8K files are saved, which correspond to the four banks, 0,1,2,and3. Each filename is prefixed by the number of the bank from which it was saved. Example: a cartridge with the given filename CARTRIDGE will be saved in four parts as follows:

```
    0CARTRIDGE     1CARTRIDGE     2CARTRIDGE     3CARTRIDGE
```

When you reload a previously saved cartridge, note that you should enter the name only. Do not add the numerical prefix. This will be done automatically by the program. Remember that any data already in a bank will be destroyed when you load into that bank, so be sure to have saved it first if it is to be required later.

Individual banks may also be saved or loaded as required, in which case no number will be added to the filename. You must also specify which bank (0,1,2,3) is to be loaded or saved. The "X" option will bypass the bank switching register to allow saving of an existing 8-ROM cartridge which can be plugged in instead of Smartcart before loading Smartcart Backup. Many 8-ROM cartriges can be installed into Smartcart eg Mikro Assembler, Stack Help, Arrow. Others such as Epyx Fastload, Quickdisk,Simons Basic and game cartridges use a different bankswitching system and are not compatible.

TAPE TURBO.

Supplied with Ramdisk tape version only.

This is a stand alone tape turbo utility which can also operate independently of the Ramdisk. The program is supplied as a Ramdisk file and is present in the directory when you install the Ramdisk. To enable the tape turbo, enter:

```
LOAD "TAPE TURBO",7 (return) and RUN
```

The tape turbo sits high in the computers memory and is compatible with all basic programs and sequential files. Loading speed is increased by about 6 times for program files and about 3 times for sequential files.

IMPORTANT. All tape turbos save data to tape in a special format. Turbo data can only be loaded when the tape turbo is in memory and initialised. If you want to load programs recorded at standard speed it will be necessary to disable the tape turbo.

To disable the tape turbo enter SYS 52482 (return)
To re-enable the tape turbo enter SYS 52480 (return)

If you press RUN/RESTORE or press the cartridge reset button the turbo wll be disabled. Enter the above SYS number to re-enable the turbo.

Load and initialise the tape turbo before using Smartcart Backup.