ENIGMA DISC IMAGER

THE MOST ADVANCED DISC ROM AVAILABLE FOR THE BBC MICRO

```
Enigma DISC Imager 0.2

READ    <dr> <tr> <sec> <no sec> <add>
WRITE   <dr> <tr> <sec> <no sec> <add>
SREAD   <dr> <tr> <sec> <no sec> <add>
SWRITE  <dr> <tr> <sec> <no sec> <add>
DEDIT   (dr) (tr)
SEDIT   (dr) (tr)
IDEDIT  (dr) (tr)
DBACK   <S dr> <D dr> (80,40) (0,1)
DCOPY   <S dr> <D dr>
RDIR    <dr> <S dir> <D dir> (G,S)
DISS    <add> (offset add)
ROMID   (addr)
MOVE    <dest add> <sou add> <len>
TULOCK
TLOCK
TTOD    <dr> (speed)
DTOT    <dr> (speed)
FORMAT  <dr> <trks> (DV)
SFORM   <dr> <tr#> <trID> <sec> <size> <sec#>
DUALFORM <dr> <trks>
VERIFY  <drv> (D)
FREE    (drv) (H)
REPAIR  <dr> <tr> <tr>
ENTER   <fsp> <sec> <len> (xeq) (load)
MEDIT   (add)
OSBYTE  <p1> <p2> as in *FX p1,p2
BSHOW   (OFF) or (out ch.)
WSHOW   (OFF) or (out ch.)
```

Altra Roms

209 North St., Leeds, LS7 2AA.
Tele. (0532 451508)

*HELP ENIGMA

```
Enigma DISC Imager 1.07
  BSHOW  <on/off>(p)
  CLR
  DBACK  <Sdr><Ddr>(40/80)(0/1)
  DCOPY  <Sdr><Ddr>
  DEDIT  (dr)(t/s)
  DFIND  <string>(t/s)(dr)
  DISS   <addr>(offset)(@rom)
  DTAPE  <dr>(300)
  DUALFORM <dr><trks>
  ENTER  <:dr.dir.fsp><sect><len>
  FORMAT <dr><trks>(DVS)
  FREE   (H)(dr)
  IDDUMP (dr)(40/80)or(@trk)
  IDEDIT <dr><trk>
  MEDIT  (addr)
  MOVE   <dest><srce><fin/+ext>
  OSBYTE <p1><p2> as *FX p1,p2
  RDIR   <dr><Sdir><Ddir>(S/G)
  READ   <dr><t/s><sects><addr>
  REPAIR <dr><trk>
  ROMID  (page)(addr)(on/off)
  SEDIT  (dr)(trk)(sect)(40/80)
  SFORM  <dr><trk><trkid><sects><size>
  SHIFT  <srce><dest><ext>
  SREAD  <dr><trk>(addr)
  SWAP   (dr)
  TDISC  (300)
  TLOCK  <on/off>
  TULOCK <on/off>
  TXCOPY
  VERIFY <dr>(trks)(D)
  WRITE  <dr><t/s><sects><addr>
  WSHOW  <on/off>(p)

OS 1.20
>
```

ALTRA ENIGMA DISC IMAGER

A ROM containing a set of utilities designed to make the most of the single density disc system used by Acorn, together with some for general purpose, such as a disassembler.

Contents are shown by *HELP ENIGMA (or *H.E.)

Commands:
All commands are entered as *NAME, the name may be in upper or lower case and abbreviated, ie: *FORMAT, *FO. or *for. will all run the disc formatter.

The minimum abbreviation that can be used for any particular command depends on which other ROMs are fitted to the computer.

If other ROMs have commands with similar names, Altra ENIGMA commands may be optionally prefixed with a capital 'A', ie: *AFORMAT, *Afor. etc. (Alternatively *ROMID can be used to switch other ROMs 'off').

Compatibility:
All the disc utilities are compatible with the current Acorn disc filing systems (DFS 0.9X and 1.X). Other DFS makes cannot be guaranteed, although the Watford DFS has been tested and appears to be compatible.

Command parameters:
Most of the utilities take one or more parameters following the command name. Parameters must be separated from the command name, and from each other, by at least one space. Those parameters that are optional are shown in the instructions and by *HELP, enclosed in parentheses (). Compulsary parameters are shown in angle brackets <>.

Most of the parameters are required in hexadecimal, exceptions are shown in the individual instructions, (ie. number of tracks in *FORMAT).

Some parameters are shown as <t/s>. This means that either an absolute sector number (in hex), or track/sector can be entered, ie: 1E2 or 30/2 will both access the same sector. Standard Acorn tracks contain 10 sectors, numbered 0 to 9.

M. Yell and I. Wilson 1984
Altra Roms
209 North Street,
Leeds 7. ©.

*BSHOW

Osbyte Show

SYNTAX: *BSHOW <ON/OFF> (output stream)

Displays all calls made to OSBYTE. The contents of A, X and Y registers are displayed on entry as well as exit.

EXAMPLE

```
>*FX 200,3

OSBYTE with
On Entry  A=C8 X=03 Y=00
On Exit   A=C8 X=00 Y=00
```

Options

(output stream)

This value is entered in decimal or hex (hex preceded by &). The value determines the output stream to which all OSBYTE displays are sent. The value of this parameter corresponds to that used with *FX 3. Therefore if one wanted printout to printer only, a value of 10 or &A would be appropriate. The default setting is to screen only.

Note: This routine uses the OSBYTE vector at &20A and also the ROM vector table on page &D. Use is also made of the stack for variable storage and locations &9E and &9F on zero page. Any overwriting of these locations will either cause the routine to stop or may hang the machine.

*CLR

Memory Clear

SYNTAX: *CLR

Clears the BBC's memory from &400 to &8000.

*DBACK

Special Disc Backup

SYNTAX: *DBACK <source drive> <dest. drive> (40/80) (0/1)

Intelligent disc backup routine. Backup discs with non standard formats. Copes with deleted data formats, non standard sector sizes and tracks with multiple sector sizes. In the event of source discs with unformatted tracks, the destination disc need NOT be a virgin unformatted disc as the routine will "unformat" the tracks concerned.

Options

40/80 Selection of disc size
The backup always defaults to 40 tracks.

0/1 Option of pressing the space bar between source and destination drive.

0 Off
1 On

This option can be used when copying from drive 0 on one disc to drive 2 on a second disc and being able to press the space bar between switching discs.

The options must be quoted in strict order.

Note: The memory limitations imposed by the BBC necessitates the use of all the machine's memory as well as the memory in pages &9, &A and &C. The user is therefore reminded that all character definitions will be lost, and that it is advised not to use the serial interface while the routine is in operation.

*DCOPY

Selective Disc File Copier

SYNTAX: *DCOPY <source drive> (destination drive)

Allows the user to selectively copy files from one drive to another. Files are selected by placing the cursor at the appropriate file and pressing either Copy to make a selection, or Delete to reverse the decision. The files are copied in the order in which they are selected. This order is indicated by the number in parenthesis next to the file name. The user then presses Tab to start the copying.

KEYS

↑	cursor up
↓	cursor down
←	cursor left
→	cursor right
Copy	select file for copying
Delete	deselect file
Tab	initialise copying

Note: The memory limitations imposed by the BBC necessitates the use of all the machine's memory as well as the memory in pages &9, &A and &C. The user is therefore reminded that all character definitions will be lost, and that it is advised not to use the serial interface while the routine is in operation.

*DEDIT

Disc Sector Editor

SYNTAX: *DEDIT (drive) (sector) (tracks)

NORMAL ACORN DISC FORMAT ONLY

Allows individual sectors to be read from disc, displayed and edited in hex and ASCII, then written back.

If the drive is not given the current drive is used.
If the sector number (in hex) is not given, the editor will start at sector 0.
If the number of tracks on the disc is not given, the number will be read from the catalogue. Therefore if the catalogue sectors are corrupt or incorrect, or the disc is not standard Acorn format, the number of tracks should be given, in decimal, with drive and sector number.

DISPLAY MODE:
is 40 or 80 columns, depending on the screen mode on entry:- Modes 0,1,2,3 will select 80 columns in mode 3, modes 4,5,6,7 select 40 columns in mode 7. 40 column mode is useful for editing catalogue sectors or with TV displays. 80 column mode shows an entire sector on the screen without scrolling.

On entry the start sector is read in and displayed. The cursor will be on the hex side, byte 00, and shown as a flashing block. This is EDIT mode.

EDIT Mode KEYS

TAB	Move cursor between hex and ASCII sides of the screen.
CURSOR Keys ↑ ↓ ← →	Move cursor around the current side of the screen, hex or ASCII. If necessary the screen scrolls up or down.
SHIFT ↑	Move cursor up 16 lines.
SHIFT ↓	Move cursor down 16 lines.
SHIFT-COPY	Copy entire screen to printer.
CTRL ←	Move back 1 sector.
CTRL →	Move forward 1 sector.
CTRL ↑	Move in 1 track.
CTRL ↓	Move out 1 track.
RETURN	Enter Command mode.
ESCAPE	Return to original language or application, no write performed.

EDITING:

ASCII	Any character with ASCII code 0 to &7E can be entered from the keyboard. Use Ctrl for characters 0-1F. (Ctrl M inserts &D, the Return key exits the editor).
Hex.	Use keys 0-9 and A-F to edit individual nybbles.

WRITING:
If a sector has been edited, any attempt to change sector (Ctrl cursors or Return), will give the prompt: "Write-back ?". If you wish to write the sector back press "Y", any other key will cause the pending sector change to be executed.

The prompt "Write sector#? (CR=origin)" will appear. Return will write the sector back to its original sector. If you wish to write it elsewhere enter either:

The absolute hex sector number (CR), ie: 12F or,
Track/Sector (CR), both in hex, ie: 30/3

Confirmation is then required to the prompt: "Write T/S to t/s, OK?". Any key except "Y" will return to the "Write sector#?" prompt. If successfully written, the pending sector change will be executed. If the write fails, the error is shown on the error line at the top of the screen and the prompt "Try again ?" appears. Only "Y" will re-attempt the write.

COMMAND Mode
The cursor appears normal on the bottom line, with the prompt: "SECTOR ? <CR=next>".

COMMAND Mode KEYS

RETURN	Read next sector.
COPY	Repeat the last function, ie re-read the current sector, or go back to write mode, or if entered from *DFIND continue searching.
SHIFT-Cursors	Change sector or track, as for Edit mode.
ESCAPE	Exit disc editor.

Alternatively an absolute sector number of Track/Sector number can be given, ie:
12F would read sector &12F
1F/9 would read track &1F, sector 9.

No check is made on the sector number in track/sector type input, allowing non standard sector numbers to be read/written. Note that the next/previous sector commands and absolute sector# assume normal Acorn format, sectors numbered 0-9.

ERRORS

If an error occurs during a read or write, the error is reported at the top of the screen, and command mode is entered. The system tries 3 times before reporting an error. If the error is "fatal", ie sector not found, no data will be displayed for a read. "COPY" can be used to try the function again if required.

If "deleted data" is encountered, the choice of writing the sector back as normal or deleted is offered.

Example screen dump of DEDIT:

```
Track: 06   Sector: 01   Drive: 0

Abs.Sector: 03D  Error:OK

       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

00    54 41 59  D 20 52 54 53  D  D  D 5C 2A 2A 2A 2A        TAY. RTS...\****
10    2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A        ****************
20    2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A        ****************
30    2A 2A 2A  D 2E 50 52 4D 45 53 32 20 4A 53 52 20        ***..PRMES2 JSR
40    4F 53 57 52 43 48  D 20 49 4E 59  D 20 4A 4D 50        OSWRCH. INY. JMP
50    20 50 52 4D 45 53 31  D 2E 50 52 42 4C 4E 4B 20         PRMES1..PRBLNK
60    43 4D 50 20 23 30  D 20 42 45 51 20 50 52 42 4C        CMP #0. BEQ PRBL
70    4B 32  D 20 54 41 58  D 20 4C 44 41 20 23 33 32        K2. TAX. LDA #32
80     D 2E 50 52 42 4C 4B 31 20 4A 53 52 20 4F 53 57        ..PRBLK1 JSR OSW
90    52 43 48  D 20 44 45 58  D 20 42 4E 45 20 50 52        RCH. DEX. BNE PR
A0    42 4C 4B 31  D 2E 50 52 42 4C 4B 32 20 52 54 53        BLK1..PRBLK2 RTS
B0     D 5C 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A        .\**************
C0    2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A        ****************
D0    2A 2A 2A 2A 2A 2A 2A 2A 2A  D 2E 52 45 4C 20 4C        *********..REL L
E0    44 41 20 23 22 26 22  D 20 4A 53 52 20 4F 53 57        DA #"&". JSR OSW
F0    52 43 48  D 20 4C 44 59 20 23 31  D 20 4A 53 52        RCH. LDY #1. JSR
```

*DFIND

Search disc for specified character sequence

SYNTAX: *DFIND “string” (t/s) (drive)

NORMAL ACORN DISC FORMAT ONLY

Searches the disc for any occurrence of the specified string. On finding a match the disc sector editor is entered with the cursor on the first character of the string. Normal disc editing may then be done, (see *DEDIT), or the search continued by pressing Return.

If the sector is edited, Copy will continue the search, while entering a track/sector number or using the Ctrl/cursors will stay in the normal disc editor.

The ”string” must be enclosed in quotes, and the normal OS system of using ¦ and ! to insert control characters and quotes may be used, (as in *KEY etc.)

The search normally starts at sector 0, but the start sector may be given as an option in t/s format, ie:
*DFIND “fred” E/6 or *DFIND “fred” 92
would start a search for ”fred” at track &E, sector 6.

The current logged on drive is the default drive for the search, another drive may be optionally specified after a track/sector number.

Note: In order to increase the speed, sectors are not searched in a fixed order, therefore sectors may be presented for editing out of numeric order.

Use MODE 7 to maximise the search speed.

*DISS

Disassembler

SYNTAX: *DISS <address> (offset) (@rom)

Disassembles memory to screen or printer. Each line shows the address, opcode bytes, ASCII equivalent characters and the instruction mnemonic. All calls to the OS are labelled, ie: JSR OSWRCH etc.

Example:

```
80A1  F0 22     ."   BEQ &80C5
80A3  B9 05 0F  ...  LDA &0F05,Y
80A6  C9 40     .@   CMP #&40
80A8  D0 F6     ..   BNE &80A0
80AA  20 EE FF   ..  JSR OSWRCH
```

***DISS <address>**

Disassembles memory starting at the given hex address, ie: *DISS D00.

***DISS <address> <offset>**

Disassembles memory starting at 'address', but relative addresses are given as if the code started at 'offset'. This allows code to be disassembled from a position it does not normally run from, ie: *DISS 3000 8000 could disassemble a ROM image residing at &3000 but make the addresses appear as if the code were actually at &8000.

***DISS <address> <@page>**

Selects the given ROM page when the address is the region &8000 to &BFFF, allowing the disassembly of a resident sideways ROM, ie:
*DISS 8000 @C
will disassemble the ROM in page C.

The disassembler is entered in 'step mode', where a key must be pressed between each instruction. Pressing 'S' toggles in and out of 'continuous mode', allowing unattended disassembly to a printer etc. Other keys are also active in step mode, viz:

S	Toggle between step and continuous modes.
Ctrl B/C	Printer on/off.
Ctrl N/O	Paged mode on/off.
Escape	Return to input: a new address, offset or ROM page can be entered, syntax as for initial entry.

*DTAPE

Selective Disc File to Tape Dump

SYNTAX: *DTAPE <source drive> (300/1200)

Allows the user to selectively dump files from a disc to tape. Files are selected by placing the cursor at the appropriate file and pressing either Copy to make a selection, or Delete to reverse the decision. The files are copied in the order in which they are selected. This order is indicated by the number in parenthesis next to the file name. The user then presses Tab to start the copying.

Options

The tape baud rate defaults to 1200 Baud, and 300 can be selected by entering 300.

KEYS

↑	cursor up
↓	cursor down
←	cursor left
→	cursor right
Copy	select file for copying
Delete	deselect file
Tab	initialise copying

Note: The memory limitations imposed by the BBC necessitates the use of all the machine's memory as well as the memory in pages &9, &A and &C. The user is therefore reminded that all character definitions will be lost, and that it is advised not to use the serial interface while the routine is in operation.

*DUALFORM

Combined 40/80 track disc formatter

SYNTAX: *DUALFORM <drive> <tracks>

NORMAL ACORN DISC FORMAT ONLY

Formats a disc so that it can be used in 40 and 80 track drives. The disc must be formatted in an 80 track drive. The maximum number of tracks is 20.

FORMAT:

Track 0 is always formatted.

Tracks <tracks> to <tracks> x 2 - 1 (ie: 20 to 39 for tracks = 20), are formatted for 80 track.

Tracks <tracks> x 2 to <tracks> x 4 - 2 (ie: 40 to 78) are formatted for 40 track, numbered 20 - 39, by double stepping. The unformatted tracks (1 to 19) are locked out by a dummy file.

Any files must be written in both 40 and 80 track modes, for the disc to be readable in either type of drive.

The system cannot be guaranteed, especially as some older 40 track drive heads cannot read 80 track data.

*ENTER

Creates a disc directory entry without saving anything

SYNTAX: *ENTER <:dr.dir.fsp> <sector> <length> (exec addr)
(load addr)

NORMAL ACORN DISC FORMAT ONLY

Allows deleted files to be recovered or alternative names for the same file to be generated.

<:dr.dir.fsp> The complete filename including drive and directory.

<sector> The start sector of the required file in hex, as given by *INFO or *FREE H.

<length> The number of bytes in the file in hex.

All 3 parameters above must be given, the following two are optional:

(exec addr) Execution address of the file, if any, in hex.

(load addr) Load address for file.

All the parameters required are as obtained from *INFO.

Example: *ENTER :0.$.FILENAM 7A 3026

would create an entry for a file called FILENAM on drive 0 in directory $. The start sector is &7A and length is &3026. As execution and load addresses are not given they will be written as 0.

Note: No check is made for duplicate names or clashing sectors.

Errors:

Catalogue full – no room left for another entry.

Bad attribute – Sector < 2 or greater than the number on the disc, or missing parameter or bad hex.

*FORMAT

Disc Formatter

SYNTAX: *FORMAT <drive> <tracks> (DVS)

***FORMAT <drive> <tracks>**

Formats the disc in the named drive with the given number of tracks. Both drive number and tracks MUST be given (in decimal). Any number of tracks may be formatted. (Normally 40 or 80).

***FORMAT <drive> <tracks> D**

Format the disc on both sides, eg:

*FORMAT 1 80 D

will format the disc in drive 1 for 80 tracks on both sides.

If the disc is not new, the message —
"Already formatted, ok to reformat?"
will appear, only Y (or y) is accepted as yes, any other key will cause the format to be abandoned before writing to the disc.

The disc is normally verified as it is formatted, and the track numbers displayed (in hex). A "?" after a track number means the track had to be reread/reformatted before it passed as OK, and may therefore be suspect.

***FORMAT <drive> <tracks> V**

Format without verifying.
To quickly reformat a disc known to be good, the normal verify can be defeated by adding a 'V'.

***FORMAT <drive> <tracks> S**

Format the disc with special catalogues for use with *SWAP. These allow up to 60 filenames per side.

As FORMAT uses no user RAM space outside the normal disc buffers, a disc may be formatted at any time without overwriting any program or text (Wordwise/View etc.)

Errors: Drive numbers other than 0 - 3 will cause "Bad Drive". Tracks less than 1 or more than 255 will cause "Bad track#".

Note: Uses bytes &A0 to &AB in page 0 and all page &F (normal sector buffer for Acorn DFS series 0.9x and 1.x).

*FREE

Disc free space calculator

SYNTAX: *FREE (H) (drive)

***FREE <drive>**

NORMAL ACORN DISC FORMAT ONLY

Calculates the amount of free space remaining on the disc in ‘drive’.
(if not given the current drive is assumed).

ie: *FREE 2

DISC-TITLE (age) <tracks>
Free gaps: 4 12 31
47 Sectors free (11 Kbytes)
Max gap 31 sectors (8 Kbytes)
5 Catalogue entries free

‘Free gaps’ shows the number of unused sectors (if any) in blocks between files.

‘Max gap’ is the largest contiguous space available without compacting the disc.

***FREE H**

As well as reporting the size of any gaps it also reports their position by start sector, both in hex.
Order is start sector then size.

ie: *FREE H

DISC-TITLE (age) <tracks>
Free gaps:
 02A 004
 06E 00C
 109 01F
 2F sectors free… etc.

Errors: If the catalogue sectors are unreadable, automatically verifies track 0 and reports.

*IDDUMP

Track ID Display

SYNTAX: *IDDUMP (drive) (40/80) option 1
*IDDUMP <drive> <@track> option 2

Produces a dump of the disc's track and sector ID's. The size of the disc drive (40/80) can be specified, the default value is 40 tracks.

The following information is presented:

Tr.#	....	Physical track number	(hex)
No.S	....	The number of sectors on the track	(hex)
Sec.#	....	The physical sector number	(hex)
Tr.ID	....	The sector track ID	(hex)
SecID	....	The sector ID	(hex)
Head#	....	The sector head number	(hex)
IDsiz	....	The size of the sector as given in the ID information	(hex)
REsiz	....	The actual real size of the sector as it can be read	(hex)
Error	....	The reporting function which reports if a track is unformatted, or if a sector is deleted data… etc.	

Options

Option 1 gives a dump of the whole disc. The default setting is 40 tracks.

Option 2 gives a dump of just one track. The track number must be given in hex.

Technical Notes:

The program uses the BBC's memory up to &22FF. This means it can only be used in a BASIC program if the setting of PAGE is &2300 or greater. Because the command uses routines from *DBACK, use is also made of the memory in PAGEs &9, &A and &C. The user is therefore reminded that all character definitions will be lost, and that it is not advised to use the serial interface while this routine is in operation.

*IDEDIT

Track ID Editor

SYNTAX: *IDEDIT <drive> <track>

Allows editing of the sector ID information of a track. The ID information is displayed on the screen in the following way:

P Sec .... The physical sector number (hex)

TrkID .... The sector track ID (hex)

Head# .... The sector head number (hex)

SecID .... The sector ID (hex)

Size .... The size of the sector:

0 ..... 128 bytes
1 ..... 256 bytes
2 ..... 512 bytes
3 ..... 1024 bytes
4 ..... 2048 bytes

KEYS

↑ cursor up

↓ cursor down

← cursor left

→ cursor right

Tab write back.

Shift Copy dump screen to printer.

Editing is done with the cursor keys and only hex input is accepted. On completion the Tab key must be pressed for completion of the editing and a prompt is given before writing back. When the track is written back, the track is reformatted for the new ID's before the track data is written back.

Note: Care must be taken in that if all the track ID's are not the same, the track data cannot be written back or reread again. If the track ID's are not the same, a warning message will be displayed before the prompt to write back.

*MEDIT

Memory Editor

SYNTAX: *MEDIT (addr)

Displays memory in both hex and ASCII. Bytes may be edited in RAM using either hex or ASCII.

DISPLAY MODE:

is 40 or 80 columns, depending on the screen mode on entry:- Modes 0,1,2,3 will select 80 columns in mode 3, modes 4,5,6,7 select 40 columns in mode 7. The cursor in either mode is a flashing block.

The display will start at the given hex address, if any. If no address is given the current value of OSHWM is used, (default BASIC PAGE).

KEYS:

TAB — Move cursor between hex and ASCII sides of the screen.

CURSOR Keys ↑ ↓ ← →

Move cursor around the current side of the screen, hex or ASCII. If necessary the screen scrolls up or down.

SHIFT ↑ — Move cursor up 16 lines.

SHIFT ↓ — Move cursor down 16 lines.

SHIFT-COPY

Copy entire screen to printer.

RETURN or
ESCAPE — Return to original language or application.

EDITING:

ASCII — Any character with ASCII code 0 to &7E can be entered from the keyboard. Use Ctrl for characters 0-1F. (Ctrl M inserts &D, the Return key exits the editor).

Hex — Use keys 0-9 and A-F to edit individual nybbles.

*MOVE

MOVE a (BASIC) program, or block of memory

This command has several forms:

***MOVE**

Will move the current BASIC program to the default PAGE value of &E00. The values of PAGE, LOMEM and TOP are adjusted to suit.

***MOVE <page>**

Moves the current program to the given value of <page>. The value is rounded down to the nearest page boundary, ie:

*MOVE 1130

will move the program to PAGE = &1100. The values of PAGE, LOMEM and TOP are adjusted to suit.

If there is insufficient room between <page> and HIMEM for the program, the program will not be moved and an error message "Bad Address" given.

***MOVE <destination> <source> <finish>**

Moves the block of memory between source and finish to the given destination address, ie:

*MOVE 34F0 2A16 2F34

will move the area of memory between &2A16 and &2F34 up to &34F0. Note that this is exclusive of the finish address itself (as in *SAVE etc.)

***MOVE <destination> <source> <+extent>**

In this case the number of bytes to be moved is given by the extent. ie:

*MOVE 34F0 1A16 +1C3

will move &1C3 bytes from &1A16 up to &34F0.

There is no restriction on any of the addresses in the last two forms of MOVE. All the moves are "intelligent", such that the source memory is not overwritten before being moved.

*OSBYTE

Osbyte Intercept

SYNTAX: *OSBYTE <parameter 1> <parameter 2>
as in *FX parameter 1, parameter 2.

Allows user priority of OSBYTE, and so enables the user to override any calls made to OSBYTE.

For example: if one wishes to override OSBYTE 200, so that the machine's memory is not cleared on break, one could type *OSBYTE 200 0. Any *FX 200,2 or *FX 200,3 call that is made would be overwritten to become *FX 200,0.

Note: This routine uses the OSBYTE vector at &20A and also the ROM vector table on page &D. Use is also made of the stack for variable storage and locations &9E and &9F on zero page. Any overwriting of these locations will either cause the routine to stop or may hang the machine.

*RDIR

Global or selective renaming of directory

SYNTAX: *RDIR <drive> <source directory> <dest. directory> (G/S)

This allows the global or selective renaming of disc directories. For example, if we wished to do a global rename of directory D to E on drive 1, we would enter:

*RDIR 1 D E or
*RDIR 1 D E G

Options

If the selective option is chosen (S), then a prompt Y/N is given before the file information is changed.

*READ

*WRITE

Read or Write specified sectors to/from memory

SYNTAX: *READ <drive> <t/s> <sectors> <addr>
*WRITE <drive> <t/s> <sectors> <addr> (D)

NORMAL ACORN DISC FORMAT ONLY

Read or Writes sectors directly, without the need for filenames.

Any number of sectors may be transferred between disc and memory, starting at the given sector.

<drive>	Drive number, must be given.
<t/s>	Sector (or Track/Sector) to start the transfer from or to.
<sectors>	Number of sectors to be transferred, in hex. Only complete sectors can be transferred, one sector = 256, bytes.
<addr>	Address (hex) in memory where data is to be transferred to or from.
(D)	Option for Write only. Writes data as 'Deleted Data'.

EXAMPLE:

*READ 1 F/6 20 3000
Read drive 1, starting at track &F, sector 6, for &20 sectors, load to &3000 in memory.

*WRITE 0 2E 5 2A00
Write to drive 0, starting at sector &2E, for 5 sectors, from &2A00 in memory.

*REPAIR

Reformats / Rewrites a single track

SYNTAX: *REPAIR <drive> <track>

NORMAL ACORN DISC FORMAT ONLY

If a disc fails to verify this will attempt the following:

1: Read as many sectors from the given track as possible into memory.
2: Prompt for confirmation.
3: Reformat the track.
4: Write the original data back.

Both the drive number and track number MUST be given.

The track number is given in hex, as reported by VERIFY. Multiple tracks may be repaired by adding further track numbers, ie:
*REPAIR 0 1C 3E 1A

Data is recovered from sectors that verify with Data CRC errors, however some of the bytes may be corrupted.
Sectors that are totally unreadable are written back as all zeros.

The sector data is read into memory from OSHWM upwards, thus it will overwrite any BASIC programs or text in RAM at the time.

If the disc is physically damaged and fails to reformat, the data cannot be written back, and should be *SAVEd to a new disc if required.
(OSHWM is the default value of BASIC PAGE, ie:
*SAVE TRACK 1900 +A00).

Example:

```
*REPAIR 0 1C
Track 1C, sector 0,  OK
                     Data saved
Track 1C, sector 1,  OK
                     Data saved
Track 1C, sector 2,  Data CRC error
                     Data saved
Track 1C, sector 3,  OK
                     Data saved
Track 1C, sector 4,  OK
                     Data saved
Track 1C, sector 5,  IDCRC error
                     Data lost
Track 1C, sector 6,  Sector not found
                     Data lost
Track 1C, sector 7,  OK
                     Data saved
Track 1C, sector 8,  OK
                     Data saved
Track 1C, sector 9,  OK
                     Data saved

Repair ?
```

*ROMID

ROM Identification

SYNTAX: *ROMID (page) (addr) (ON/OFF)

Shows the page number, version number, title and copyright string for each sideways ROM fitted in the BBC.
eg. *ROMID

F-01 BASIC (C)1982 Acorn
E-03 DFS (C)
D-03 Altra PROBE 4.01 (C)1984 IJW
...
0-07 Altra Enigma Disc Imager 1.07 (C)1985 MDY/IJW

***ROMID <page> <address>**

To copy a ROM image down to RAM, add the required ROM page number (in hex) and address (also hex).

ie: *ROMID F 3000

will copy the ROM in page F down to &3000. A command line consisting of the ROM title and addresses is displayed, which can then be copied to save the image, viz:

*SAVE BASIC 3000 +4000 D9CD 8000

The execution address given is the OS restart, so images reloaded into ROM emulators will automatically be validated.

***ROMID :drive**

Checks every file on the disc in the named drive for valid ROM images, listing the Version, Title and Copyright string of those found.
ie: *ROMID :1

1 04 Altra PROBE 4.01 (C)1984 IJW
2 01 BASIC (C)1982 Acorn
3 03 DFS (C)
Load No.?

Entering a file number (the numbers in the left column), will then load that file to &8000.

***ROMID <page> ON/OFF**

Disables or re-enables the ROM in the given page socket. BASIC cannot be disabled by this means. ROMs which use the main vectors (filing systems etc), will still respond to the standard OS commands, *LOAD etc, but not to their own unique * commands. ROMs currently ‘on’ are tagged with ‘-’ in *ROMID listings.

*SEDIT

Special Sector Editor

SYNTAX: *SEDIT (drive) (track) (sector) (40/80)

Allows individual sectors to be read from disc, displayed and edited in hex and ASCII, then written back. This editor provides for discs with non standard formats and intelligently edits deleted data formats, and non standard sector sizes.

If the drive is not given, the current drive is used.
If the track number (in hex) is not given, the editor will start at physical track 0.
If the sector number (in hex) is not given, the editor will start at physical sector 0.
If the number of tracks on the disc is not given (in decimal 40/80), the default number of tracks is 40.
If any parameters are quoted they must be used in strict order.

DISPLAY MODE:
is 40 or 80 columns, depending on the screen mode on entry:- Modes 0,1,2,3 will select 80 columns in mode 3, modes 4,5,6,7 select 40 columns in mode 7. 40 column mode is useful for editing catalogue sectors or with TV displays. 80 column mode shows an entire 256 byte sector on the screen without scrolling.

On entry the start sector is read in and displayed. The cursor will be on the hex side, byte 00, and shown as a flashing block. This is EDIT mode.

EDIT Mode KEYS

TAB	Move cursor between hex and ASCII sides of the screen.
CURSOR Keys ↑ ↓ ← →	Move cursor around the current side of the screen, hex or ASCII. If necessary the screen scrolls up or down.
SHIFT ↑	Move cursor up 16 lines.
SHIFT ↓	Move cursor down 16 lines.
SHIFT-COPY	Copy entire screen to printer.
CTRL ←	Move back 1 sector.
CTRL →	Move forward 1 sector.
CTRL ↑	Move in 1 track.
CTRL ↓	Move out 1 track.
RETURN	Enter Command mode.
ESCAPE	Return to original language or application, no write performed.

EDITING:

ASCII	Any character with ASCII code 0 to &7E can be entered from the keyboard. Use Ctrl for characters 0-1F. (CtrlM inserts &D, the Return key exits the editor).
Hex	Use keys 0-9 and A-F to edit individual nybbles.

WRITING

If a sector has been edited, any attempt to change sector (Ctrl cursors or Return), will give the prompt: "Write back ?". If you wish to write the sector back press "Y", any other key will cause the pending sector change to be executed.

If the write fails, the error is shown on the error line at the top of the screen and the prompt "Try again ?" appears. Only "Y" will re-attempt the write.

COMMAND Mode

The cursor appears normal on the bottom line, with the prompt: "SECTOR ? <CR=next>".

COMMAND Mode KEYS

RETURN	Read next sector.
COPY	Repeat the last function, ie reread the current sector, or go back to write mode.
SHIFT-Cursors	Change sector or track, as for Edit mode.
ESCAPE	Exit disc editor.

Alternatively an absolute sector number or Track/Sector number can be given, ie:
12F would read sector &12F
1F/9 would read track &1F, sector 9.

No check is made on the sector number in track/sector type input, allowing non standard sector numbers to be read/written.

ERRORS

If an error occurs during a read or write, the error is reported at the top of the screen, and command mode is entered. If the error is "fatal", ie sector not found, no data will be displayed for a read. "COPY" can be used to try the function again if required.

Example screen dump from SEDIT:

```
Drive: 00  No. of Sectors: 0A
Sector Size: &0100 Bytes
Phys. Track No: 04  Phys. Sect. No: 00
Track ID. No  : 24  Sector ID No  : F6
ERROR : Deleted data

00    16  4 1C  2 11  F 10 17  0  6 1F  0  0  0  0  0        ................
10     0 17  0  C  C  0  0  0  0  0  0 17  0  D  0  0        ................
20     0  0  0  0  0 17  0  1 20  0  0  0  0  0  0 17        ........ .......
30     0  2 2D  0  0  0  0  0  0 17  0  A 20  0  0  0        ..-......... ...
40     0  0  0  1  1  0 6F F8  4  1  8  8 FE  0 FF 7E        ......o........~
50    2C  2  1  E EE FF 2C 20 32  6  1  0 FE 78 7E  3        ,.....,  2....x~.
60     1  1 FF FD 11 20 80  1  0  0 FF  1  1  4  1  4        ..... ..........
70    F8 2C  4  6  8 16  0  0 81 7E  0 20 72 1B A9 90        .,.......~. r...
80    A2 FF 20 26 1D A9  0 85 70 A9 19 85 71 A0  0 B1        .. &....p...q...
90    70 20 EE FF C8 C0 43 D0 F6 20 89 1B A9 10 A2  3        p ....C.. ......
A0    20 F4 FF A9 60 8D 32  2 A9  2 8D 25  2 A9 32 8D         ...£.2....%..2.
B0    24  2 A9 BE A2  8 20 26 1D A9 C8 A2  3 20 26 1D        $..... &..... &.
C0    A9  D A2  0 20 26 1D A9 E1 A2 80 20 26 1D A9  C        .... &..... &...
D0    A2  0 20 26 1D A9  D A2  2 20 26 1D A9  4 A2  1        .. &..... &.....
E0    20 26 1D A9  9 A2  0 20 26 1D 20 E2 1C A9  0 85         &..... &. .....
F0    70 A9 11 85 71 A9 62 85 72 A9 29 85 73 20 2C 1D        p...q.b.r.).s ,.
```

*SFORM

Format a track to any given sector format

SYNTAX: *SFORM <drive> <track no.> <track id> <sectors> <size>
(start sector)

Allows non Acorn standard tracks to be formatted.

<drive> Drive number, must be specified.

<track no.> Physical track number (hex). No check is made for disc size, so beware of using tracks which your drive does not support.

<track id> The number (hex) which is to be written out as the track identification, which may or may not be the same as the physical track number.

<sectors> The number of sectors (in hex) required on the track, range 1 to &12.

<size> Code for size of sectors required, 0-4.

The size is coded as follows:

Code	Bytes	Max sectors/track
0	128	18
1	256	10
2	512	5
3	1024	2
4	2048	1

Using an illegal size/sectors combination will give "Bad attribute".

(start sect.) By default sectors are numbered 0,1,2, ... on each track. They may be given any sequence (0-FF) by optionally specifying the start number.
VERIFICATION:
The track is verified after formatting as the normal *VERIFY cannot be used later.
If the sectors are numbered such that they roll over zero, ie: FE, FF, 0, 1, 2 ...,
the verify will fail, even though the track may be correctly written.

EXAMPLE:
*SFORM 0 7 A3 10 0 1
Format drive 0, track 7: Write as track &A3 with 16 sectors of size 0, (128 bytes), numbered 1, 2, 3 ...

*SHIFT

Move a block of memory

SYNTAX: *SHIFT <source> <destination> <extent>

Similar to *MOVE, but with a different syntax for compatibility with other systems.

Moves the number of bytes given by ‘extent’ from ‘source’ to ‘destination’.

<source> Source address in hex.

<destination>
Destination address in hex.

<extent> Number of bytes to move, in hex.

Eg: *SHIFT 1900 E00 236A

Moves a block of memory of &236A bytes length, from &1900 down to &E00.

*SREAD

Special Track Reader

SYNTAX: *SREAD <drive> <track> (memory address)

Reads non standard tracks with non standard sector sizes or deleted data. The initial memory address to which the track data is loaded is printed out as well as the finish address.

For those who wish to incorporate this routine in a program, the memory pointer for Fmem+1 (Final address + 1) is located in &74 (l.s.b.) and &75 (m.s.b.)

Sample Printout:

```
*SREAD 0 4 3000

Imem.   :&3000
Fmem+1 :&3A00
>
```

Options

If an address is not specified the default address to which data is loaded is OSHWM

Note: Because the command uses routines in *DBACK use is made of the memory in pages &9, &A and &C. The user is therefore reminded that all character definitions will be lost, and that it is not advised to use the serial interface while this routine is in operation.

*SWAP

Swap catalogues on a dual catalogue (60 file) disc

SYNTAX: *SWAP (drive)

SWAP can only be used on discs formatted with special catalogues by *FORMAT (S). These allow up to 60 files per side by treating each side as two logical halves, each being half the true size of the disc. Each half has a separate catalogue holding 31 filenames, but one is used to hold the alternative catalogue, leaving 30 files per half.

The dummy file appears in each catalogue as !.!!!!!!! L, and is locked (L). This file must not be deleted, otherwise the other half of the disc will be lost.

*SWAP physically swaps the catalogue sectors around, thus the disc must not be write protected for it to work.

Using *SWAP on a standard format disc without the special catalogues will give the error 'File not found'.

COPYING: *COPY will work normally, except that files cannot be directly copied between catalogues on the same side of a disc.

BACKUP: *BACKUP can only be used if the current catalogue holds the true disc size, use *FREE to check.

*TDISC

Automatic Tape to Disc Transfer

SYNTAX: *TDISC (300)

Transfers all files from a tape to disc automatically. The only operator intervention required is to rewind the tape in the case of any load errors.

Default baud rate is 1200, 300 may be selected by using *TDISC 300. Files are written to the current drive.

As the Tape filing system allows filenames longer than that accepted by the Disc system, some filenames may be truncated. If the second character of the tape filename is a period '.', the first character will become the directory for the disc filename. Normally files will be saved in the current directory.

Examples:

Tape	Disc
ABCDEFGHIJ	$.ABCDEFG
R.PROGNAME	R.PROGNAM

Only standard Acorn format tape files can be transferred. (Also see *TULOCK).

Note that many cassette programs or games will not run directly in a disc based machine, due to the changes in memory allocation between the filing systems. The tape load/execution addresses are transferred to the disc, and may have to be changed.

*TLOCK

Tape Locker

SYNTAX: *TLOCK <ON/OFF>

Locks cassette based programs. The programs can then only be *RUN.

Notes: Uses the EVENT vector at &220 and the ROM vector table on page &D. Any overwriting of these locations will either cause the routine to stop or may hang the machine.

*TULOCK

Tape Unlocker

SYNTAX: *TULOCK <ON/OFF>

Unlocks cassette based programs. This is very useful if used in association with *TDISC.

Notes: Uses the EVENT vector at &220 and the ROM vector table on page &D. Any overwriting of these locations will either cause the routine to stop or may hang the machine.

*TXCOPY

Copy text screen to printer

SYNTAX: *TXCOPY

Copies the text in the current text window to a printer (any make). If no window is set then the whole screen is copied.

User defined characters in modes 0 to 6 are printed as spaces, thus allowing any printer to be used.

*VERIFY

Disc Verifier

SYNTAX: *VERIFY (drive) (tracks) (D)

***VERIFY**

NORMAL ACORN DISC FORMAT ONLY

Verifies the complete disc in the current drive.

All sectors on the disc are checked for legibility. As each track is verified its number is shown (in hex). Any faults are reported with track and sector number, ie:

```
>*VERIFY

Verifying 0
00  01  02  03  04  05  06
Track 6, Sector 2, Data CRC error
07  08  09  0A  0B  0C  0D  0E  0F  10  11
Track 11, Sector 8, Sector not found
12  13  14  15  16  17  18  19  1A  1B  1C  1D  1E  1F  20  21  22
23  24  25  26? 27

Check complete
```

A "?" after a track number shows the track had to be re-read before passing as ok. (Up to 3 rereads are done before reporting an error.)

***VERIFY <drive>**

The drive to be verified may be given as a parameter, ie:

```
*VERIFY 2
```

will verify the disc in drive 2.

***VERIFY <drive> <tracks>**

The number of tracks to verify may optionally be given (in decimal), in this case the drive number MUST be given as well. ie:

```
*VERIFY 1 10
```

would verify the first 10 tracks of drive 1.

If the catalogue sectors are unreadable and the number of tracks have not been specified, a 40 track disc is assumed.

***VERIFY <drive> D**

Both sides of a disc (in a double sided drive) can be verified by adding 'D'.
Escape will work at any time during the verification.

*WSHOW

Osword Show

SYNTAX: *WSHOW <ON/OFF> (output stream)

Displays all calls made to OSWORD. The contents of A, X and Y registers are displayed on entry as well as exit. A 13 byte parameter block is also displayed in hex. It is up to the user to interpret the parameter block.

EXAMPLE:

```
>SOUND 0,-15,10,10

OSWORD with
On Entry  A=07 X=37 Y=00
Parameter Block
00 00 F1 FF 0A 00 0A 00 00 00 00 00 00
On Exit   A=07 X=04 Y=F1
```

Options

(output stream)

This value is entered in decimal or hex (hex preceded by &). The value determines the output stream to which all OSWORD displays are sent. The value of this parameter corresponds to that used with *FX 3. Therefore if one wanted printout to printer only, a value of 10 or &A would be appropriate. The default setting is to screen only.

Note: This routine uses the OSWORD vector at &20C and also the ROM vector table on page &D. Use is also made of the stack for variable storage and locations &9E and &9F on zero page. Any overwriting of these locations will either cause the routine to stop or may hang the machine.