

# SOFTWARE SERVICE

***Floppy-wise***

**PLUS**

**THE  
ULTIMATE  
BBC MICRO  
UTILITY**

## FITTING INSTRUCTIONS

- 1 Turn off and unplug the computer.
- 2 Remove the four screws marked 'fix'.
- 3 Remove lid.
- 4 Unscrew the two or three screws holding the keyboard in place and pull it slightly towards you.
- 5 In the right hand nearest corner there are five sockets; the FLOPPY-WISE PLUS ROM is to be inserted in one of the vacant positions.
- 6 With the notch on the rom furthest away from you push the rom into the socket.
- 7 Check visually that all the pins are inserted into the socket and that the rom notch is towards the back of the machine.
- 8 Refix keyboard and lid.
- 9 Plug in and switch on the computer.
- 10 The normal

BBC Computer 32K

Floppy-wise PLUS

(DFS etc.)

BASIC

>

should be displayed.

- 11 Type \*H.F. and press RETURN.
- 12 The list of commands as shown on the back cover should be displayed.

FLOPPY-WISE PLUS is now installed.

# **Floppy-wise PLUS**

---

## **1. INTRODUCTION.**

Floppy-wise PLUS is an extremely useful utility for the BBC Microcomputer which extends its range and allows a number of new functions to be performed in conjunction with disc drives and an Acorn compatible DFS.

The program is incorporated in an EPROM which is inserted in one of the free sideways ROM positions or in an extension board.

This means that the utilities are always available to the user without having to load into the machine when required. Also they do not use up valuable user memory within the machine.

Once the device has been fitted in the machine, typing \*HELP FLOPPY. will list all the commands and their syntax.

The following section gives a brief description of the utilities.

**NOTE:** All commands may be prefixed by the letter "S" to avoid confusion with similarly named commands in other sideways firmware.

## 2. BRIEF DESCRIPTION OF UTILITIES.

### ASCII (<str>)

This prints a "Help" screen for all ASCII values.

### AUTOSAVE (OFF)(<fsp>)

This will automatically save a file to disk every 4 minutes while the program is being developed. This will prevent accidental loss of data which could occur due to power failure or accidental operation of the Break key.

### BAD

Used to repair a Basic program which has been corrupted in some way and cannot be listed.

### BMOVE (<dest pg>)(<srce pg>)

This will move a Basic program from its present position to the specified position in memory.

### CLEAR

This will clear all variables from the machine.

### CLM

This clears all of the memory .

**CLONE <s drv><d drv><s trk><e trk>**

This allows backup copies to be made of all protected disks. This is an extremely fast machine code utility and will copy a fully protected 40 track disk in under 3 minutes.

NOTE: It is illegal to make backup copies of copyright material other than for your own personal use.

**CONVERT (<drv>)**

This allows a 40 track disk to be automatically converted to 80 tracks.

**CROM (<ROM No.>)(<fsp>)**

Used to copy a resident ROM to disk.

**DEERRORS**

This prints up a list of the Disk Error codes.

**DOWNLOAD <fsp>(<adr>)**

This will load a file from the current filing medium and move it to the specified address.

**DSEARCH <str> (<drv>)(<trk>)(<trks>)**

This will search the disk for the specified string.

**DSPACE (<drv>)**

This details all space available on the current (or specified) disk without the need to compact it.

**DTAPE (<fsp>)**

This takes a file from disk and transfers it on to cassette.



**DTS <drv><fsp>**

This takes the total contents of a disk and transfers them to cassette for archive storage.

**DZAP (<drv>)(<trk>)(<trks>)**

This will display a sector of a disk and allow editing if desired.

**EDIT (<KEY No.>)**

To read and edit the function keys.

**FCOMPARE <fsp>(<adr>)(<len>)**

This compares computer memory with a file on disk.

**FIND <str>**

This searches for a string within a BASIC program.

**FORMAT <drv><trks>**

Used to format disks before they can store data.

**FREE**

This displays the unused memory available for Basic programs, and the current position of Lowmem, Hmem, Page and Top.

**IDZAP (<drv>)(<trk>)(<len>)(<trks>)**

This allows the sector ID's on a track to be edited or formatted with the specified sector ID's identities.

**NCOPY <s drv><d drv>**

This allows selected files from the source disk to be transferred to the destination disk.

**MENU (<drv>)**

This provides an automatic menu to run programs on disk with single key operation.

**MOVE <adr><adr><len>(S)**

This allows the specified block of memory to be moved.

**MSEARCH <str>(<adr>)(S)**

This will search memory for the specified string.

**MZAP (<adr>)(S)**

This displays the area of memory from the specified address for inspection and possible editing.

**PREPARE <drv><trks>**

This command is used on a "clean" disk which is required to be protected, before any files have been stored.

**PROTECT (<drv>)**

This command is used to finally copy-proof a disk.

**RAMLOAD <fsp><RAM posn.>**

To load a program into sideways RAM.

**RCOPY <s drv><d drv><fsp><fsp>**

This allows a file to be transferred and renamed at the same time.

**RECOVER <trk><sct><sct><adr><drv>**

This allows the specified sector(s) to be loaded directly from disk into memory.

**RESTORE <trk><sct><sct><adr><drv>**

This allows memory to be stored directly on disk at the specified place.

**ROMS(<ROM no.>)(<E/D>)**

This shows the status of all the sideways firmware present in the machine and allows them to be switched in and out.

**RSEARCH <str>(<ROM No.>)**

Allows a ROM to be searched for an occurrence of a given string.

**RZAP (<ROM No.>)**

This allows the specified sideways ROM to be inspected.

**TDISK (<fsp>)**

This will automatically transfer the specified file(s) from tape to disk.

**TDS <drv><fsp>**

This reverses the DTS command and returns the contents of the prepared cassette to disk.

**TOKENS**

This will print a 'Help' table of all the Basic tokens.

**VERIFY (<drv>)**

This checks that a disk has been formatted correctly.



## SYNTAX

The following is a list of the arguments used to describe the syntax of commands.

<trk>	track
<len>	length
<adr>	address
<sct>	sector number
<s drv>	source drive
<d drv>	destination drive
<fsp>	filename
<str>	string
<s trk>	start track
<e trk>	end track
<drv>	drive
<trks>	tracks on disk
<RAM posn>	RAM position

The following represent optional inputs.

(<str>)	string
(<fsp>)	filename
(OFF)	cancel command
(<drv>)	drive
(<trk>)	track number
(<trks>)	number of tracks
(<ROM no.>)	ROM number
(<dest pg>)	destination page
(<srce pg>)	source page
(<KEY No.>)	function key number
(<len>)	length in hex
(<adr>)	address to be used
(S)	second processor
(<E/D>)	Enable/Disable

## USING ARGUMENTS

1. All arguments specified must be separated by one or more spaces or commas.

2. <srce pg> and <dest pg> are entered in HEX without prefixing with '&'. If they are not specified at a page boundary, the value will be automatically changed to the page value below. e.g. 1234 will become 1200. 123 will become 0100.

3. <addr> and <len> are entered in HEX as above. The value is used as specified and should be in the range 0000-FFFF. (Leading zeros may be omitted.)

4. <ROM No.> and <RAM posn> are entered in HEX in the range 0-F.

5. All other numeric arguments are entered as a DECIMAL value.

6. <str> may contain up to 127 characters but must be enclosed within quotes if spaces are to be included. HEX and DECIMAL values may also be included by inserting them within double bars. e.g. 255 or &FF. The '&' must be specified before HEX values. As quotes and double bars have a special purpose, these can only be entered in a string by using their HEX or DECIMAL values. (&22 for quotes and &7C for double bars.)

## ASCII

**FUNCTION:** To indicate the ASCII value of a specified character.

**SYNTAX:** \*ASCII (<str>)

**SHORTEST ABBREVIATION:** \*AS.

### DESCRIPTION:

It is often necessary to know the ASCII value of a certain character or string of characters. Use of this command will print the required value(s) on the screen.

### EXAMPLE:

\*ASCII \_ will print a full list of codes in table format.

\*ASCII STRING \_ will print a list of all ASCII values corresponding to the whole string.

**NOTE:** "Paged Mode" is automatically entered and, if necessary, the SHIFT key should be pressed to scroll the screen.

## AUTOSAVE

**FUNCTION:** To save a copy of the current Basic program in memory every 4 minutes to prevent accidental loss of data.

**SYNTAX:** \*AUTOSAVE (OFF)((fsp))

**SHORTEST ABBREVIATION:** \*AU.

### DESCRIPTION:

While working on programs if a power failure occurs or the Break key is inadvertently pressed all data can be lost. It is therefore advisable to regularly save files during the course of program development. This utility does an automatic save every 4 minutes. The operation is done transparently and will not interrupt the input of data to the program. The program is intelligent and will not allow deletion of a previously saved file of the same name as the file specified in the command. Any attempt to do this will result in your being given a choice of continuing or aborting the operation and selecting another file name.

**NOTE:** \*AUTOSAVE OFF will switch the utility off.

**NOTE:** \*AUTOSAVE will give the current state of this option.

## **BAD**

**FUNCTION:** To repair a Basic program which has been corrupted and cannot be listed.

**SYNTAX:** \*BAD

**SHORTEST ABBREVIATION:** \*BAD

### **DESCRIPTION:**

Sometimes a program in memory can be corrupted. This can be done in a number of ways by altering memory locations or by pressing BREAK without OLD. Normally, under these conditions, the program is not listable. The command \*BAD will search the computer's memory and try to repair the program. Usually all of the program can be recovered, but sometimes it is only possible to repair part of it.

## **BMOVE**

**FUNCTION:** To move a Basic program in memory.

**SYNTAX:** \*BMOVE (<dest pg>)(<srce pg>)

**SHORTEST ABBREVIATION:** \*BM.

### **DESCRIPTION:**

This is similar to DOWNLOAD except that it will move a program already in memory.

If the destination page is omitted, the program will be moved to page &E00.

If the source address is omitted, the current value of PAGE is assumed.

Cassette mode is automatically entered if the destination address is between &C00 and &1100.

## CLEAR

**FUNCTION:** To clear all variables from the machine.

**SYNTAX:** \*CLEAR

**SHORTEST ABBREVIATION:** \*CLE.

**DESCRIPTION:**

After running a Basic program, some memory will be set aside for variable storage. This will clear these variables from the machine's memory and set the resident variables to zero.



## CLM

**FUNCTION:** To clear all memory in the computer.

**SYNTAX:** \*CLM

**SHORTEST ABBREVIATION:** \*CLM

**DESCRIPTION:**

This command will simply wipe out any code which is in the computer memory, leaving it completely clear. It is identical to using the \*FX200,3 command and then pressing BREAK.

## CLONE

**FUNCTION:** To make backup copies of protected disks. This is an extremely fast machine code program and a fully protected 40(80) track disk can be transferred in less than 3(6) minutes.

**SYNTAX:** \*CLONE <s drv><d drv><s trk><e trk>

**SHORTEST ABBREVIATION:** \*CLO.

### DESCRIPTION:

Some commercial programs have software protection incorporated on the disks so that they cannot be copied. It is always advisable to make backup copies of all software in case the files become accidentally corrupted or damaged in any way. This allows copies of your own to be made, so that you can safely store away the originals for use in an emergency.

### ARGUMENTS:

<s drv> \_ Source drive number.  
<d drv> \_ Destination drive number.  
<s trk> \_ Start track number.  
<e trk> \_ End track number.

**EXAMPLE:** \*CLO. 0 1 0 39 \_ will backup the first 40 tracks of the disk present in drive 0 onto the disk in drive 1.

continued overleaf....

CLONE continued...

NOTE: If only one disk drive is present in the system, the first 2 arguments should be designated "0" and you will be prompted to insert the appropriate disk.

NOTE: It is illegal to make backup copies of copyright material other than for your own personal use.

NOTE: Track numbers may contain 1 or 2 digits.

NOTE: ESCAPE may be pressed at any time. You will then be given the option to continue or abort the operation.

## CONVERT

**FUNCTION:** To create an 80 track disc from a 40 track disc while retaining all data on the disc.

**SYNTAX:** \*CONVERT (<drv>)

**SHORTEST ABBREVIATION:** \*CON.

### DESCRIPTION:

Should you have an 80 track disc drive and a number of 40 track discs these are wasteful in storage capacity.

It is possible to format another disc and copy all the files from the 40 track disc onto it and then reformat the 40 track disc to 80 tracks. This utility allows you to do all this in one operation - on the same disc - without removal of the disc. Simply insert the 40 track disc, wait until the job has completed, and remove the reformatted 80 track one.

**NOTE:** This should not be attempted on protected discs, that is discs that will not \*VERIFY, as this may corrupt the disks.

## **CROM**

**FUNCTION:** To copy any ROM resident in the machine to disc.

**SYNTAX:** \*CROM (<ROM No.>)(<fsp>)

**SHORTEST ABBREVIATION:** \*CR.

### **DESCRIPTION:**

This command will simply copy the contents of the specified ROM to disc.

If the last argument is omitted, you will be prompted to enter a filename.

If the first argument is omitted, a ROM menu will be displayed and you will be able to select a ROM from the list given, after which, you will be prompted for a filename under which it will be saved.

This command sets up the load and execution addresses so that if the file is to be loaded into sideways RAM all that has to be performed is a \*RUN of the file. This will then load the ROM based software into the currently selected RAM slot and then perform a BREAK command to enable the ROM to be recognised.

## **DERRORS**

**FUNCTION:** To indicate the faults associated with the Disk Error numbers.

**SYNTAX:** \*DERRORS

**SHORTEST ABBREVIATION:** \*DE.

**DESCRIPTION:**

If a Disk Error occurs while trying to use a disk you can type \*DE. to give an indication of what is wrong with the disk.

## DOWNLOAD

**FUNCTION:** To load and relocate a program in memory.

**SYNTAX:** \*DOWNLOAD <fsp>(<adr>)

**SHORTEST ABBREVIATION:** \*DO.

### DESCRIPTION:

The specified file is loaded into the computer and then relocated at the specified address. This is usually to reclaim memory used by the DFS in order to allow programs sufficient memory space in which to run.

If an address is not specified the file's own load address is used.

If the address is within the range &C00-&1100, the cassette filing system is selected.



## DSEARCH

**FUNCTION:** To search a disk for the occurrence of the specified string.

**SYNTAX:** \*DSEARCH <str>(<drv>)(<trk>)(<trks>)

**SHORTEST ABBREVIATION:** \*DSE.

### DESCRIPTION:

Using this command a disk can be searched for the occurrence of any specified string. The disk is scanned in track and sector order, beginning at the specified track. On finding the string, the Disk Edit menu will be entered and the appropriate sector displayed. The data may then be edited (see DZAP command) or the search can be continued to the next appearance by pressing the COPY key.

**NOTE:** This command will work with Acorn compatible or double density disks.

**NOTE:** This command will not work with non-standard disks.

## DSPACE

**FUNCTION:** To check the available free space on a disk without compacting the disk.

**SYNTAX:** \*DSPACE (<drv>)

**SHORTEST ABBREVIATION:** \*DSP.

## DTAPE

**FUNCTION:** To transfer files from disk to tape.

**SYNTAX:** \*DTAPE (<fsp>)

**SHORTEST ABBREVIATION:** \*DT.

**DESCRIPTION:**

If it is required to transfer files from Disk to Cassette, this command can be used to do it automatically. If no argument is used, all files on the current disk will be transferred.

**EXAMPLE:** \*DTAPE File1 will transfer the file "File1" to cassette.

**NOTE:** If the drive and library are not specified in the filename, then the default drive and library are used.

**NOTE:** If all files are to be transferred, the files are copied regardless of library.

## **DTS**

**FUNCTION:** To archive the total contents of a disk to cassette as a backup copy.

**SYNTAX:** \*DTS <drv><fsp>

**SHORTEST ABBREVIATION:** \*DTS

### **DESCRIPTION:**

This allows the whole of a disk (including catalogue) to be archived on cassette as a backup in case of loss or damage to a disk. The filename specified may contain up to 8 characters and will be suffixed with the current track number as the disk is archived.

**EXAMPLE:** \*DTS 0 DISK will give a cassette filename of DISK00, DISK01, etc as the disk is copied.

**NOTE:** This command is compatible with Acorn standard and double density disks.

**NOTE:** This command will not work with non-standard disks.

## DZAP

**FUNCTION:** To allow interactive disk editing.

**SYNTAX:** \*DZAP ((drv))((trk))((trks))

**SHORTEST ABBREVIATION:** \*DZ.

### DESCRIPTION:

This is an extremely powerful disk editor. Disks are displayed a sector at a time (all sized sectors are catered for) and the sectors can be edited and rewritten to the disk in their original positions.

The disc will be scanned sequentially as required by reference to the initial command or by use of the curser keys - see commands below. Logical and physical track and sector numbers are permanently displayed at the top of the screen along with the drive number.

If a bad track or sector is encountered, a disk error number is displayed under the track and sector numbers.

Operation of the ESCAPE key will exit from the edit screen. If, however, any data on the screen has been altered, you will be given the option of saving any edited changes or leaving the disk unaltered. A 'Y' answer to the question will result in the disk itself being altered. Any other key depression will abort the operation.

**EDITOR COMMANDS:**

- <^> Move UP one line.
- <v> Move DOWN one line.
- <➤> Move RIGHT one byte.
- <◀> Move LEFT one byte.
- <SHIFT ^> Move to the top left byte.
- <SHIFT v> Move to the bottom right byte.
- <SHIFT ➤> Move to the right-hand byte.
- <SHIFT ◀> Move to the left-hand byte.
- <CTRL ^> Move FORWARD one track.
- <CTRL v> Move BACK one track.
- <CTRL ➤> Move FORWARD one sector.
- <CTRL ◀> Move BACK one sector.
- <TAB> Change base for edit entry - toggles between ASCII and Hex.
- <ESCAPE> Exit editor.
- <A-Z,0-9> Edit in hex.
- <Any key> Edit in ascii.

## EDIT

**FUNCTION:** To display the current settings of the function keys.

**SYNTAX:** \*EDIT (<KEY No.>)

**SHORTEST ABBREVIATION:** \*E.

**DESCRIPTION:**

This is used to display the current settings of the function keys. This will then allow easy editing using BASIC's editing routines.

If a KEY number is specified, only that key will be displayed.



## **FCOMPARE**

**FUNCTION:** To compare a file on disk with the contents of memory.

**SYNTAX:** \*FCOMPARE <fsp>(<adr>)(<len>)

**SHORTEST ABBREVIATION:** \*FC.

### **DESCRIPTION:**

Sometimes it is desired to check that a file has been correctly saved. This utility will allow a comparison.

If the last two arguments are omitted, the current value of PAGE is assumed as the start address.

If an address is specified, this is taken as the start address in memory from which the comparison starts.

The last argument allows part of a file to be compared. If this is omitted, the whole file is compared.

## FIND

**FUNCTION:** To locate the occurrences of a particular string within a BASIC program.

**SYNTAX:** \*FIND (<str>)

**SHORTEST ABBREVIATION:** \*FI.

### DESCRIPTION:

It is often necessary to know the lines at which a particular string occurs within a BASIC program. This function saves the user time by finding these automatically.

### EXAMPLE:

\*FIND FRED \_ Prints the line numbers at which "FRED" occurs.

**NOTE:** BASIC Tokens can be searched for by enclosing the token value between double bars. (\*TOKENS will display these values)

## **FORMAT**

**FUNCTION:** To initialise a blank disc to prepare it to accept data.

**SYNTAX:** \*FORMAT <drv><trks>

**SHORTEST ABBREVIATION:** \*FD.

## **FREE**

**FUNCTION:** To display the available memory for Basic programs, the current size of any resident BASIC program and the current values of Lowmem, Hmem, Page and Top.

**SYNTAX:** \*FREE

**SHORTEST ABBREVIATION:** \*FR.

## IDZAP

**FUNCTION:** To edit the sector identities of the specified track.

**SYNTAX:** \*IDZAP (<drv>)(<trk>)(<len>)(<trks>)

**SHORTEST ABBREVIATION:** \*ID.

### DESCRIPTION:

This command allows sector identities on a track to be displayed and edited. This makes it possible to create a protected disk or to unprotect a protected disk.

If no arguments are specified, the current drive will be selected and track zero will be displayed. Movement between tracks is done using CTRL and CURSOR up/down keys.

The identities may be altered by using the cursor keys to move to the required position and then changing the value. When the desired values have been set up, moving to the next track or pressing ESCAPE will prompt you to save the identities to disk. If 'Y' is pressed, the new identities are copied to disk and the data within each sector remains the same. Actual sector lengths cannot be changed.

**NOTE:** If the track contains sectors of more than one size, the message 'NON-STANDARD TRACK' appears and the identities will not be allowed to be re-saved.

If drive and track numbers are specified, these will be used instead of the default values.

If a sector length is specified, (0-5) this will create dummy identities which may be edited

and saved to disk. No data is saved using this argument, only a blank track is created.

The final argument is used to specify the number of tracks on a disk which has a corrupt catalogue. If this value is not specified, the number of tracks on the disk is taken from the catalogue.

NOTE: Sector lengths (0-5) are defined as follows:

- 0 128 bytes
- 1 256 bytes
- 2 512 bytes
- 3 1024 bytes
- 4 2048 bytes
- 5 256 bytes for double density

## MCOPY

**FUNCTION:** To copy a pre-selected number of files from one disk to another.

**SYNTAX:** \*MCOPY<s drv><d drv>

**SHORTEST ABBREVIATION:** \*MC.

### DESCRIPTION:

Sometimes it is required to copy only some files from one disk to another. On entry into this program a menu is shown on the screen of the source disk catalogue. A cursor can be moved around the filenames (using the cursor keys) and the files to be selected indicated (with COPY key). Files can be deselected using DELETE key. RETURN will set the procedure in motion and all selected files will be copied.

**NOTE:** If a file of the same name exists on the destination disk, a prompt will be displayed asking if you wish to delete it. Pressing 'Y' will overwrite the file and continue with the copying.



## MENU

**FUNCTION:** To provide an automatic disk menu for single key entry to programs on a disk.

**SYNTAX:** \*MENU (<drv>)

**SHORTEST ABBREVIATION:** \*ME.

### DESCRIPTION:

On entering the command the automatic Menu will search the current disk for programs, which will be displayed on screen alongside a letter which can then be pressed to run the programs. Basic or Machine Code programs are catered for automatically with no necessary pre-knowledge by the user.

To set up the disk for this kind of operation it is necessary to put Basic programs into B Directory and Machine Code programs into M Directory. Programs in neither of these two directories are not displayed in the menu.

Pressing keys 0,1,2 or 3 will change drives. Operation of the key corresponding to the required program will CHAIN or \*RUN that program.

**NOTE:** It is also possible by pressing 0-3 and the BREAK key to select the disk to be searched. e.g. 3 BRK will scan drive 3. Pressing S and the BREAK key will act the same as using 0 and BREAK.

## MOVE

**FUNCTION:** To shift a block of memory from one place to another.

**SYNTAX:** \*MOVE <adr><adr><len>(S)

**SHORTEST ABBREVIATION:** \*MOV.

**DESCRIPTION:**

This enables a block of memory to be moved in memory. This is similar to the BMOVE command but will move any block of memory and not just Basic programs.

**ARGUMENTS:**

<adr> start of source address.

<adr> start of destination address.

<len> length of block to be moved.

(S) Enables memory to be moved in the second processor.

**NOTE:** This command is intelligent and will cater for an overlap in memory locations. Therefore there is no danger of losing data during the move.

## MSEARCH

**FUNCTION:** To search computer memory for occurrences of the specified string.

**SYNTAX:** \*MSEARCH <str>(<adr>)(S)

**SHORTEST ABBREVIATION:** \*MS.

### DESCRIPTION:

This will allow you to search for any specified string in memory. On finding the string the Memory Editor will be entered and the start of the string displayed. Operation of the <COPY> key will find the next occurrence of the string.

Specifying an address will start the search from this point.

Second processor memory can be viewed and edited by specifying 'S' after or instead of the memory address.

If specified without an address, the value of PAGE is taken as the start address for the search.

Values other than ASCII characters can be found by stipulating the value between two double bars. e.g. 123 for Decimal and &34 for Hex.

## MZAP

**FUNCTION:** An interactive Memory Monitor/Editor.

**SYNTAX:** \*MZAP (<adr>)(<S>)

**SHORTEST ABBREVIATION:** \*MZ.

### DESCRIPTION:

This memory editor allows any section of memory to be viewed and changed if required. The byte currently set for editing is hi-lighted in a different colour and can be edited in ASCII or Hex. (Mode changed by the <TAB> key.

Default condition for entry into the editor is the current Page address. The optional address can allow the display of any other locations. If the 'S' argument is used the Second Processor (6502) memory is displayed.

Movement around the memory is done using the cursor keys as follows:-

<➤	Move right one byte
<◀	Move left one byte
<⤴	Move back one line
<⤵	Move forward one line
SHIFT <➤	Move right to end of line
SHIFT <◀	Move left to end of line
SHIFT <⤴	Move back 256 bytes
SHIFT <⤵	Move forward 256 bytes
TAB	Toggle between ASCII and HEX
ESCAPE	Exit editor

## PREPARE

**FUNCTION:** To format a disc in a special way to prepare for protection against copying. This command will overwrite any data already on the disk. On entering this command you will be prompted with the message 'Are you sure?'. Pressing 'Y' will commence formatting of the disk. Any other key will exit the command.

**SYNTAX:** \*PREPARE <drv> <trks>

**SHORTEST ABBREVIATION:** \*PRE.

## PROTECT

**FUNCTION:** To protect a disk from being copied.

**SYNTAX:** \*PROTECT (<drv>)

**SHORTEST ABBREVIATION:** \*PRO.

**DESCRIPTION:**

If it is required that a disk is to be protected it is necessary that the disk is first formatted using the "PREPARE" command before any files are stored on it. Programs can then be loaded as necessary. A menu must then be created to cover all the files on the disk so that they can be automatically run. A !BOOT file must also be loaded onto the disk to run the menu. Finally the OPT4 option must be set. (See DFS User Guide.)

\*PROTECT will now prevent copying or reading of the disk other than by using SHIFT and BREAK to load the menu program.

**NOTE:** Experienced disk users should have no difficulty with the above procedure since use of the "shift-break" boot facility is quite fundamental.

## RAMLOAD

**FUNCTION:** To allow loading of a program into sideways RAM.

**SYNTAX:** \*RAMLOAD <fsp><RAM posn>

**SHORTEST ABBREVIATION:** \*RA.

**DESCRIPTION:**

The file specified will be loaded into the sideways RAM position selected. This command may be used within a program to load data or ROM based software into one or more RAM slots.

## RCOPY

**FUNCTION:** To enable a file to be copied to another (or the same) disk under a different name.

**SYNTAX:** \*RCOPY <s drv><d drv><fsp><fsp>

**SHORTEST ABBREVIATION:** \*RC.

**EXAMPLE:** \*RC. 0 1 ZX81 BBC-B \_ will transfer the file called ZX81 on drive 0 to drive 1 and rename it BBC-B.



## RECOVER

**FUNCTION** To load the specified sector(s) from disk into memory.

**SYNTAX:** \*RECOVER <trk><sct><sct><adr><drv>

**SHORTEST ABBREVIATION:** \*REC.

### DESCRIPTION:

Sectors can be removed directly from disk to computer using this command. There they can be manipulated and replaced on the disk using the RESTORE command. Sometimes accidentally corrupted disks can be fixed using a combination of these two commands. i.e. If a CRC error has occurred in a sector, it can be RECOVERED and then RESTORED to the same place on the disk to make a perfect repair.

### ARGUMENTS

<trk> start track to be recovered.  
<sct> start sector.  
<sct> total number of sectors.  
<adr> address in memory for loading.  
<drv> drive

## RESTORE

**FUNCTION:** To save an area of memory to specified sector(s) on a disk.

**SYNTAX:** \*RESTORE <trk><sct><sct><adr><drv>

**SHORTEST ABBREVIATION:** \*RES.

**DESCRIPTION:**

This command is the direct opposite of the RECOVER command and is mainly used in conjunction with it. For details see the previous section.

## ROMS

**FUNCTION:** To display the identity and position of sideways ROM firmware available in the machine and to allow each to be switched on or off.

**SYNTAX:** \*ROMS (<ROM no.>)(<E/D>)

**DESCRIPTION:** All sideways firmware is displayed in a menu along with its position and status. ROMs can be enabled or disabled separately.

**NOTE:** \*ROMS 13 E \_ will enable ROM 13.  
          \*ROMS 11 D \_ will disable ROM 11.  
          \*ROMS \_ displays a list of the ROMs in the machine. They can then be toggled between the disabled and enabled conditions by pressing the key corresponding to their position. Operation of the ESCAPE key will exit the menu leaving the ROM status as selected.

**NOTE:** The status of the ROMs will remain as selected after pressing BREAK but they will all be re-enabled after a CONTROL-BREAK.

**NOTE:** Sideways RAM slots not containing valid ROM software will also be displayed in the menu but cannot be enabled or disabled.

## **RSEARCH**

**FUNCTION:** To allow the search for the specified string in the specified sideways ROM.

**SYNTAX:** \*RSEARCH <str>(<ROM no>)

**SHORTEST ABBREVIATION:** \*RS.

**DESCRIPTION:**

This routine allows you to search for the specified string in a sideways ROM. If the second argument is omitted a screen menu is entered. Operation of the required key will display the required ROM (or RAM). On finding the string the editor is entered and if in RAM it can be altered. Operation of the COPY key will continue the search for the next occurrence of the string.

## RZAP

**FUNCTION:** To allow inspection of the specified sideways ROM. If in RAM form, this can be edited.

**SYNTAX:** \*RZAP (<ROM no>)

**SHORTEST ABBREVIATION:** \*RZ.

**DESCRIPTION:**

This routine allows Sideways ROMs to be viewed or edited.

If the argument is omitted a screen menu is entered. Operation of the required key will display the required ROM (or RAM). Vacant RAM positions may also be entered and edited.

**NOTE:** See MZAP for description and use of keys.

## **TDISK**

**FUNCTION:** To allow tape to disk transfer of selected files.

**SYNTAX:** \*TDISK (<fsp>)

**SHORTEST ABBREVIATION:** \*TDI.

### **DESCRIPTION:**

Any files on cassette may be transferred to disk using this command. This does not mean that they will still run correctly as the memory allocation of disk filing systems is different to the cassette system. Use of \*DOWNLOAD will normally allow the program to run.

Omitting the filename from the command will allow all files on the cassette to be transferred.

**NOTE:** Disk filenames can only contain 7 characters whereas cassette filenames can be 10 characters long. During this process, therefore, filenames may be shortened on transfer by deleting any extra characters which may be on the end of a cassette filename. Care must be taken that these shortened filenames do not correspond with disk filenames already in use, as valuable data can be overwritten.

## **TDS**

**FUNCTION:** This command is the exact opposite of the DTS command. It returns the contents of a previously archived disk, now on tape, back to disk.

**SYNTAX:** \*TDS <drv><fsp>

## TOKENS

**FUNCTION:** To indicate the values of Basic Tokens.

**SYNTAX:** \*TOKENS

**SHORTEST ABBREVIATION:** \*TO.

**DESCRIPTION:**

It is often necessary to know the value of a certain Basic Token. Use of this command will print the values of all Tokens on the screen along with their execution address and number of extra bytes required by the Basic ROM. The display is scrolled through the Tokens by use of the SHIFT key.



## VERIFY

**FUNCTION:** To check that a disk has been formatted correctly.

**SYNTAX:** \*VERIFY (<drv>)

**SHORTEST ABBREVIATION:** \*V.

Floppy-wise PLUS IS COPYRIGHT SOFTWARE SERVICES  
1985.  
ALL RIGHTS RESERVED.

Any queries or suggestions regarding this  
firmware should be addressed to:-

SOFTWARE SERVICES,  
65 SOUTH MOSSLEY HILL ROAD,  
ALLERTON,  
LIVERPOOL. L19 9BG

TEL.051 427 7894



# Floppy-wise PLUS

ASCII (<str>)  
AUTOSAVE (OFF) (<fsp>)  
BAD  
BMOVE (<dest pg>)(<srce pg>)  
CLEAR  
CLM  
CLONE <s drv><d drv><s trk><e trk>  
CONVERT (<drv>)  
CROM (<ROM No.>)(<fsp>)  
ERRORS  
DOWNLOAD <fsp>(<adr>)  
DSEARCH <str>(<drv>)(<trk>)(<trks>)  
DSPACE (<drv>)  
DTAPE (<fsp>)  
DTS <drv><fsp>  
DZAP (<drv>)(<trk>)(<trks>)  
EDIT (<KEY No.>)  
FCOMPARE <fsp>(<adr>)(<len>)  
FIND <str>  
FORMAT <drv><trks>  
FREE  
IDZAP (<drv>)(<trk>)(<len>)(<trks>)  
MCOPI <s drv><d drv>  
MENU (<drv>)  
MOVE <adr><adr><len> (S)  
MSEARCH <str>(<adr>) (S)  
MZAP (<adr>) (S)  
PREPARE <drv><trks>  
PROTECT (<drv>)  
RAMLOAD <fsp><RAM posn>  
RCOPY <s drv><d drv><fsp><fsp>  
RECOVER <trk><sct><sct><adr><adr>  
RESTORE <trk><sct><sct><adr><adr>  
ROMS (<ROM No>)(<E/D>)  
RSEARCH <str>(<ROM No>)  
RZAP (<ROM No>)  
TDISK (<fsp>)  
TDS <drv><fsp>  
TOKENS  
VERIFY (<drv>)

## Software Services

65, South Mossley Hill Road,  
Allerton, Liverpool L19 9BG.

e PLUS+Softwa  
PLUS+Software  
PLUS+Software  
LUS+Software  
US+Software S  
S+Software Se  
+Software Ser  
Software Serv  
oftware Servi  
ftware Servic  
ware Service  
ware Services  
are Services+  
re Services+F  
e Services+Fl  
Services+Flo  
Services+Flop  
ervices+Flopp  
rvices+Floppy  
vices+Floppy-  
ices+Floppy-w  
ces+Floppy-wi  
es+Floppy-wis  
s+Floppy-wise  
+Floppy-wise  
Floppy-wise F  
loppy-wise PL  
loppy-wise PLU  
ppy-wise PLUS  
py-wise PLUS+  
y-wise PLUS+S  
-wise PLUS+Sc  
wise PLUS+Soft  
ise PLUS+Soft  
se PLUS+Softw  
e PLUS+Softwa  
PLUS+Softwar  
PLUS+Software  
LUS+Software  
US+Software S  
S+Software Se  
+Software Ser  
Software Serv  
oftware Servi  
ftware Servic  
ware Service  
ware Services  
are Services+  
re Services+

# Floppy-wise PLUS

ASCII (<str>)  
 AUTOSAVE (OFF) (<fsp>)  
 BAD  
 BMOVE (<dest pg>)(<srce pg>)  
 CLEAR  
 CLM  
 CLONE <s drv><d drv><s trk><e trk>  
 CONVERT (<drv>)  
 CROM (<ROM No.>)(<fsp>)  
 DERRORS  
 DOWNLOAD (<fsp>)(<adr>)  
 DSEARCH (<str>)(<drv>)(<trk>)(<trks>)  
 DSPACE (<drv>)  
 DTAPE (<fsp>)  
 DTS <drv><fsp>  
 DZAP (<drv>)(<trk>)(<trks>)  
 EDIT (<KEY No.>)  
 FCOMPARE (<fsp>)(<adr>)(<len>)  
 FIND <str>  
 FORMAT <drv><trks>  
 FREE  
 IDZAP (<drv>)(<trk>)(<len>)(<trks>)  
 MCOPY <s drv><d drv>  
 MENU (<drv>)  
 MOVE <adr><adr><len> (S)  
 MSEARCH (<str>)(<adr>) (S)  
 MZAP (<adr>) (S)  
 PREPARE <drv><trks>  
 PROTECT (<drv>)  
 RAMLOAD (<fsp><RAM posn>  
 RCOPY <s drv><d drv><fsp><fsp>  
 RECOVER <trk><sct><sct><adr><adr>  
 RESTORE <trk><sct><sct><adr><adr>  
 ROMS (<ROM No>)(<E/D>)  
 RSEARCH (<str>)(<ROM No>)  
 RZAP (<ROM No>)  
 TDISK (<fsp>)  
 TDS <drv><fsp>  
 TOKENS  
 VERIFY (<drv>)

**Software Services**  
 65, South Mossley Hill Road,  
 Allerton, Liverpool L19 9BG.

e PLUS+Software  
 PLUS+Software  
 PLUS+Software  
 LUS+Software  
 US+Software S  
 S+Software Se  
 +Softi  
 Softw  
 oftwar  
 ftware  
 ware  
 are Se  
 re Ser  
 e Serv  
 Serv:  
 Servic  
 ervice  
 rvice  
 vices-  
 ices+f  
 ces+F  
 es+Flo  
 s+Flop  
 +Floop  
 Flopp  
 floppy-  
 oppy-  
 ppy-w  
 py-wi  
 y-wise  
 -wise  
 wise f  
 ise PL  
 se PLL  
 e PLUS  
 PLUS+  
 PLUS+  
 LUS+Sc  
 US+So  
 S+Softw  
 +Software Ser  
 Software Serv  
 oftware Servi  
 tware Service  
 ware Services  
 are Servicest  
 re Servicest

**Floppy-wise PLUS**



# SOFTWARE SERVICE

***Floppy-wise***

**PLUS**

THE  
ULTIMATE  
BBC MICRO  
UTILITY