Opus.
Opus Supplies Ltd.,

DISC MANUAL
FOR THE
BBC MICROCOMPUTER
USING
OPUS PERIPHERALS

ACORN D.F.S.

D.D.O.S.

CHALLENGER

Contents

1. How to use this manual

If you have purchased a disk interface from Opus Supplies Ltd., and if the interface has not been fitted into your computer by a dealer then look at the fitting instructions in chapter 12, read and follow them carefully. If you do not feel competent to follow the instructions then contact your local dealer for assistance.

If you have never used disc drives before you should read chapters 2 and 3 which explain how to use them, and how to look after your discs. You should then read through chapter 4, carrying out the exercises given, to help familiarise you with your new equipment.

Chapter 5 provides a detailed explanation of the commands available in the Acorn DFS, and should be read by users of Opus DDOS and CHALLENGER as these systems include all the commands provided by the Acorn DFS, (and many more besides!)

The extensions provided by Opus DDOS are explained in chapter 6, it is suggested that new users should read chapter 5 before proceding to the extra facilities provided by Opus DDOS. Users of the CHALLENGER should also read this chapter, since all facilities provided by DDOS are available to users of the CHALLENGER.

CHALLENGER ADDENDUM

You will find that most protected programmes will boot immediately on your Challenger.

However for some software: ie. ELITE, it is necessary for the disc catalogue to be loaded at E00.

To enable these programmes to load you should type the following:—

```
*ENABLE CAT (RETURN)
```

You will then be able to boot the programme as normal.

2. Disc drives and discs

Installing a disc drive unit (Acorn DFS and Opus DDOS)

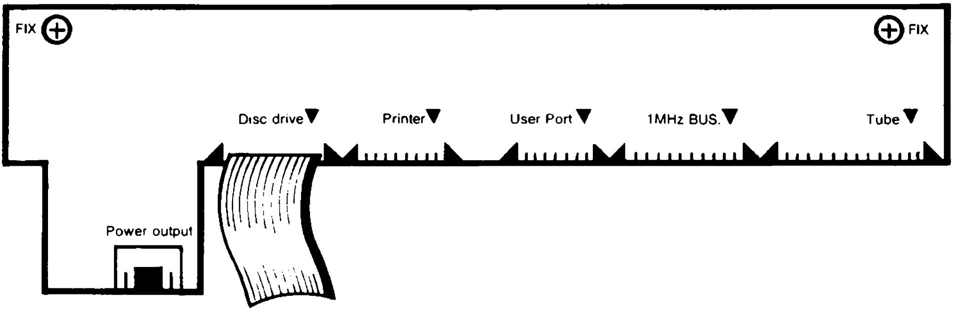
The BBC microcomputer can be fitted with one or two disc drives. When two are fitted, they are usually in the form of a dual unit in a single casing.

A disc drive is a very simple device to look at. It is a long metallic box with two cables attached to it. At the front are one or two 'letter-boxes' through which the discs will be 'posted'. The letter-boxes have doors which are opened and closed either by a lever, which turns through 90 degrees, or a flap.

There may also be an indicator light on each drive which lights when the drive is in use. At the back of the unit, there may be an on-off switch, of which more later.

Of the two cables, one is a flat ribbon cable fitted with a flat 34-way plug, which will connect to the BBC. The other is a power cable. If the disc drives have their own power supply, then you will need to fit an appropriate mains plug to this lead, with a 3-Amp fuse. If the disc drives are to be powered from the BBC, the cable will already be fitted with a 6-way connector.

To fit the drive unit, turn off and unplug the BBC. Lift up the computer, underneath you will observe a number of labelled sockets. If you have an Opus RAMDISC insert the 34-way plug into the socket marked '1MHz Bus', if you are using Acorn DFS or Opus DDOS insert the 34-way plug into the socket marked 'Disc drive'. Normally, this is possible in only one way because the plug has a small lug on top which must fit into a corresponding notch in the socket. Alternatively, the ribbon cable will have a red stripe down one side, and the plug should be aligned so that this stripe goes to the side of the socket marked by an arrow head.



If there is a 6-way power connector, this should be connected to the socket marked 'Auxiliary power socket'. It is not possible to connect this plug in other than the correct orientation. If the drive unit has its own power supply, plug in the mains power lead.

The drive unit should be positioned on a firm horizontal surface. Open the doors and remove any cardboard transit packing slips. Keep these for possible future use.

Close the doors again. Power on the BBC and the drive unit. Nothing should happen at this stage. If the indicator lights do come on or the drive makes unexpected noises, either the drive has been incorrectly connected to the BBC, or the disc filing system upgrade has been incorrectly fitted in the BBC. Turn off the BBC and the disc drive and recheck all steps, if necessary refitting the disc upgrade.

If all is quiet, type *. (RETURN) on the BBC keyboard. The drive (the upper drive of a dual drive unit) should now spin, making a constant whirr, and the indicator light should be on. If not, check the switch on the back of the drive is on. Check the mains connection or the 6-way power connector. Check the ribbon cable connection.

To stop the drive spinning, press the BREAK key. The drive unit is now installed and ready for action, but before you start using it you need to know a little about disc drives and floppy discs.

Types of disc drives

Not only do disc drives come singly or in pairs, but each drive can be either single or double sided (which means that they can use just one or both sides of a floppy disc). The numbering of the disc drives on the BBC reflects these combinations.

A disc in a single, single-sided drive is said to be on drive 0. In a single double-sided drive, the upper side of the disc is on drive 0 and the lower side on drive 2.

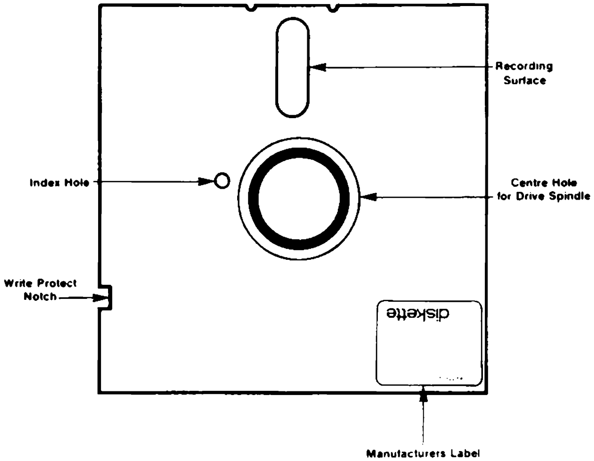
With dual drives, a disc in the upper drive is on drive 0 or 2, as above. A disc in the lower drive is on drive 1 (upper side) and drive 3 (lower side).

A disc drive can be either a 40-track or an 80-track drive, and in some cases, can be switchable between these modes (by the 40-80 switch at the back of the drive). An 80-track drive can store twice as much information on a floppy disc as can a 40-track drive.

Normally 40-track discs must be read by 40-track drives, and 80-track discs by 80-track drives. DDOS can also read 40-track discs on 80-track drives.

Floppy Discs

You will have purchased at least one box of blank floppy discs, so now is a good time to study one, in conjunction with the diagram of the floppy disc.



The disc will be in a paper or cardboard protective envelope. Whenever a disc is not being used, return it to its envelope, and then to a suitable storage box, to protect it from dust. Dust can not only destroy a floppy disc and the information on it, but it can wreck the disc drive as well.

When removed from the envelope you will see three areas of exposed magnetic recording surface; around the central hole, in a smaller offset hole called the index mark, and an elongated radial hole called the head slot. Never touch any of these surfaces, as the grease from your fingers may well ruin the disc.

Despite their name, floppy discs are reasonably robust when handled with care. However, they can easily be destroyed with heat (a carelessly placed coffee cup), by strong magnetic fields (close to a television or monitor) or by physical stress (writing onto the cover with a ball-point, for instance).

As well as having to survive all these environmental hazards, the information on floppy discs can quite easily be overwritten or deleted by human error. Because of this, it is essential that you keep at least one copy of all discs which hold information.

Discs come supplied with sticky identification labels. You should always label every disc as soon as it has any useful information on it. Put as much information as you need on the label to remind you what the disc holds. Indicate whether the disc is a master or a copy. It is better to prepare the labels before sticking them on to the disc; if you must write on a label which is already on the disc, use a felt tipped pen with no great pressure.

One way of reducing the chance of accidental loss of information is to use the write-protect tags which are supplied with the discs. They are sticky black or metallic tags. If you completely cover the small write protection notch on the side of a disc by one of these tags, a disc drive will be unable to alter the information on the disc in any way, regardless of what you tell it to do.

It should be noted here that floppy discs have as many varieties as do disc drives. Discs are described in terms of the best performance that can be expected from them. Thus the cheapest discs are described as 40-track (sometimes 48 tpi), single-sided, single density, and should not be used for double density recording, for instance.

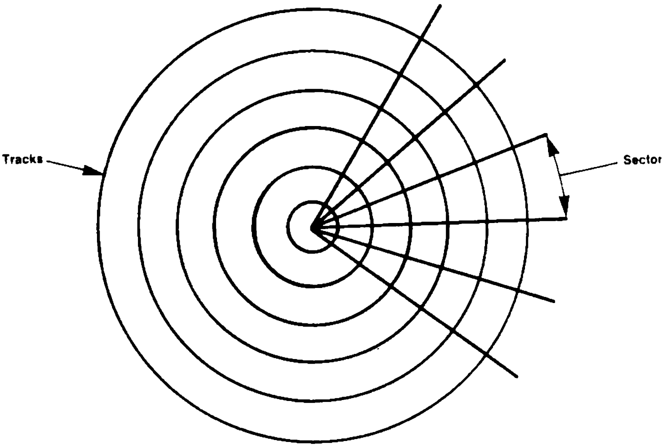
The best discs cope with 80-track (96 tpi) and double density, and are double sided. We would recommend that you always buy best quality double density discs, specified for 80-track (96 tpi) if you have 80-track drives, and double sided if you have double-sided drives. As floppy discs are relatively cheap, any other strategy is false economy.

Disc formats

A brand new floppy disc is covered on one or both sides with a uniform coating of magnetic recording material with no structure. Before such a disc can be of use it must be given structure. It must be formatted.

When formatted,a disc is divided into concentric rings called tracks. There are 40 or 80 tracks per side, depending on the disc drive.

Each track is itself divided into sectors. Standard BBC single density discs have 10 sectors per track. DDOS also offers double density format, in which each track has 18 sectors.



Each sector contains 256 data bytes, plus some control information. Thus a 40-track disc recorded in single density will hold 40*10*256 = 100K bytes of data per side, similarly an 80 track disc recorded in double density will hold 80*18*256 = 360K bytes of data per side.

As well as the basic track/sector structure, each disc is divided into volumes. A volume consists of a catalogue and one or more sectors available for data storage.

Standard single density discs consist of a single volume, whereas DDOS double density discs hold up to 8 volumes. Volumes have single letter names in the range A to H.

3 Files

A file is the basic unit of information storage in the filing system. Each volume may contain up to 31 files.

A file consists of data and a catalogue entry. The data is a collection of bytes stored in consecutive sectors on a disc. Each data byte may take any value between 0 and 255.

The catalogue entry is a collection of attributes of the file. These are the entry name, the lock, the load address, the execution address, the byte length, the allocated sector length, and the address of the first data sector.

The entry name is the primary method of access to a file.It is unique within the catalogue. It consists of a single letter directory name, a fullstop, and a filename of up to 7 characters. Neither the filename nor the directory name may contain the characters . : # *. The filing system makes no distinction between upper and lower case letters in entry names, or anywhere else for that matter.

In Acorn DFS the full file specification includes not only the entry name, but also the drive name, preceded by a colon and followed by a fullstop. Thus

```
:0.$.FRED
```

specifies the file named FRED of the directory $ on the disc in drive 0. In fact, this could probably have been typed as

```
FRED
```

because the filing system will substitute defaults for all the missing fields. Initially these defaults are drive 0, directory $ but the user may change them with the *DRIVE and *DIR commands, and the new defaults will be used until they are set again by the user, or until a reset (BREAK).

In DDOS the full file specification includes the entry name and the volume and drive names. Thus

```
:0A.$.FRED
```

specifies the file named FRED of the directory $ in volume A on the disc in drive 0. The default volume is initially volume A, and may be changed by the user with the *DRIVE command.

Returning to the catalogue attributes, the lock is an on-off switch. When the lock is on, a file cannot be altered in any way; neither the data nor the catalogue entry can be modified or destroyed. The file can still be read.

The load and execution addresses are used by various file system services concerned with manipulating whole files.

The byte length of a file is the number of data bytes, which may be zero. The allocated sector length of a file is the number of sectors set aside for the file on disc. Normally it is just sufficient to store the data bytes, and it can be derived from the byte length:

```
sector length = (byte length + 255) MOD 256
```

The address of the first data sector is used by the filing system to access the data.

Opening and Closing Files

There are several file system services which are concerned with manipulating a file as a single entity. However, when access to individual bytes is required, a file must be accessed through a channel.

A channel can be attached to a single file; the file has then been opened. To close a file is to detach it from the channel.

Certain additional attributes are associated with accessing a file through a channel. These are the channel number, the write permit, the pointer and the buffer.

The channel number identifies a particular channel. Acorn DFS and DDOS make five channels available, numbered 17 to 21. Other filing systems, such as the cassette filing system, use a different range of channel numbers.

The write permit is an on-off switch. When the switch is on, data can be written to, and read from the channel. When the switch is off, the channel can only be read.

The setting of the write permit is determined by the open mode. If the file is opened for input, the write permit flag will be off. If a file is opened for output or update, the write permit flag will be on.

More than one channel can be attached to a given file, so long as they all have their write permit switch off i.e. so long as they are all open for input.

When a file is to be opened for input or update, the file must already exist. When a file is to be opened for output, a new file of zero length is created, replacing any existing file.

When a file is opened for update, or is opened for output and replaces an existing file, the allocated sector length is initially set to that of the existing file. When a file opened for output does not replace an existing file, the allocated sector length starts at 64.

The pointer is an index of an opened file's data bytes. When a file is opened, the pointer for the channel is set to zero. The byte read or written to a channel is the byte indexed by the pointer.

The pointer is automatically incremented by one after each byte is read or written. The user may also set the pointer explicitly.

If the pointer is greater than or equal to the file length, an attempt to read a byte will fail with an end-of-file (EOF) condition. If instead a byte is written to the file, the byte length of the file is increased by one. If the file is then too long for the allocated sector length, the file is extended by adding to it as many free sectors as possible. This will fail if the file has already reached the end of the volume or the start of another file's data sectors.

The buffer is a 256-byte area in memory in which is held a copy of the sector from which the last byte was read, or to which the last byte was written. If the next byte to be read or written is in the same sector, the filing system need not actually transfer the sector to or from disc, but can use the buffer copy instead. This considerably speeds up access to files.

Of course, when a new sector is required, the filing system will have to flush the buffer (that is, update the old sector to disc if it has been changed), and then read the new sector into the buffer. The filing system will also have to flush the buffer when the file is closed, and the user may explicitly flush a buffer at any time.

Sequential Access and Random Access

A file is sequentially accessed when it is read or written from the first to the last byte in order. Sequential file access does not require the explicit use of the pointer, as the pointer is automatically incremented after each access. Sequential access is the most efficient way to exploit the buffer.

Any other mode of access is random access. The pointer is used to explicitly control which byte is accessed. The data bytes may be accessed in any order, possibly more than once or not at all. A random access file is similar to a single dimension BASIC array, except that a random access file can be extended beyond its original size.

Random access is appropriate when the next byte to be accessed is determined by the input from the keyboard. A typical random access program would be the implementation of a spreadsheet.

Fast as disc access is, badly-designed random access files can be very inefficient to access. Thus in the spreadsheet example, if the file were divided into columns, and then each column into rows, the program would run orders of magnitude slower than if the file were divided into rows and then columns. This is because a spreadsheet will normally compute and display all the results for a given row before moving onto the next.

File Types

The filing system does not distinguish between sequential and random access files. Indeed the same file can be accessed in both ways. More generally, the filing system does not distinguish between the different uses made of files. It considers a file to be a collection of data bytes, and some catalogue attributes.

However, a file is usually useful only if a particular interpretation is placed on the value of its data bytes. The appropriate interpretation for a given file is known as the file type, of which there are a few standards.

A text file is a sequential file of printable characters and cursor movements such as space, tab, backspace and newline in the ASCII character code.

Such files are commonly known as 'ASCII files', but this is less accurate than 'text files' since the term 'ASCII' refers to a particular interpretation of ANY value in the range 0 to 127, which includes some characters which are not printable on the BBC computer.

A Basic program file is a sequential file of BASIC tokens and text that represents a Basic program. Tokens are described in chapter 43 of the User Guide.

A Basic data file is a sequential file of Basic data items (integers, reals, and strings) generated using the PRINT# Basic statement. Such a file can be read using the INPUT# Basic statement. A random access Basic data file also requires the use of the PTR# Basic statement.

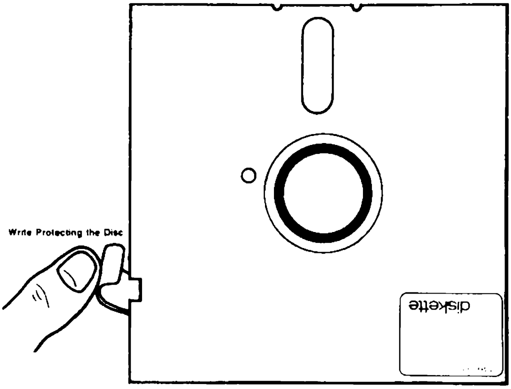
A Binary program file is a sequential file of machine code instructions and/or binary data that, when loaded into memory, forms a working machine code program, or some part of one.

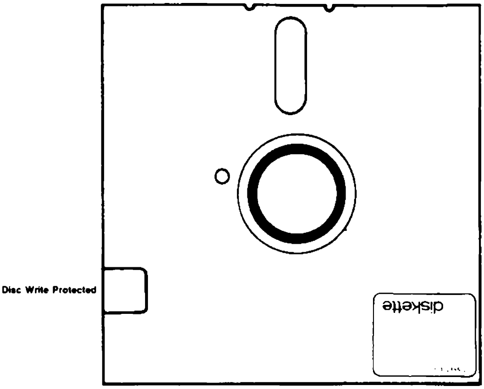
A binary data file describes any file other than one of these defined types. Its interpretation depends on the programs that access it. In Basic, a binary data file is generated using the BPUT# statement and read using the BGET# statement. A binary data file can be interpreted sequentially or as a random access file.

4 Getting Started

Formatting

Before you can store any information on a disc it is necessary to format your new discs. With Acorn DFS the program for formatting a disc is not present in the ROM which was installed in your BBC when the disc interface was fitted, and must be loaded from disc. Opus DDOS and CHALLENGER both contain the formatting program in the ROM installed in your BBC with the disc interface. If you are using Acorn DFS take your master utility disc which was supplied with your disc drive, and ensure that it has a write protect label on it. If it does not have a write protect label on it, put one on, you do not want to accidentally erase your only master disc!





Insert the master disc in drive 0, this is the top drive of a dual system. If your disc drive is a 40-80 switchable, ensure the 40-80 switch is set to the 80-track position. Type in the command:

```
*FORMAT
```

Press the RETURN key at the end of the line. The disc drive will whirr and the indicator light will come on. In a moment the formatting program will be loaded and will ask you some questions. Remove the master disc at this point.

If you have Opus DDOS or CHALLENGER you do not need to insert a master disc, just type the command as above.

The formatting program will ask you which drive you wish to format the disc in, use drive 0 in this case, and how many tracks you wish to format. If you are using an 80-track drive select 80, if you are using a 40-track drive select 40. If you are using DDOS or CHALLENGER you will also be asked which density you wish to use, select single density in this case.

When you have answered these questions insert a new disc in drive 0 and when the formatting program says 'are you sure' answer 'y'. The formatting program asks for this confirmation because when you format a disc all previous information is erased.

Saving and Loading BASIC programs

Whenever a computer language is used on the BBC, it uses the current filing system for saving and loading programs, and for all file handling. A BBC which does not have a disc interface fitted uses the tape filing system to load and save programs. When a disc interface is fitted the disc filing system is automatically made the current filing system, and all the commands which were used with the tape filing system can still be used, but now programs are loaded and stored to disc and files are opened to disc. It is not necessary to change your programs provided that you have used file names which are also legal disc file names.

Type NEW to clear the BBC of any previous programs, and then type in the following program.

```
10 CLS
20 PRINT "Programs are quicker from disc"
30 END
```

Save the program on your newly formatted disc, calling it "FIRST" by placing the disc in drive 0 and typing

```
SAVE "FIRST"
```

The indicator light on drive 0 will come on and the program will be saved onto disc. You can now check that it is there by typing

```
*CAT
```

which will show you the files on the disc. Remove the disc from your disc drive and switch off the BBC; you should never switch your BBC off or on with discs in the disc drive, or the discs can become corrupted. Switch the BBC back on and insert the disc into the disc drive. To reload the program "FIRST" type

```
LOAD "FIRST"
```

You may now type LIST to see that it has reloaded properly.

You can see that the statements in BASIC for handling files on disc are the same as those used with tape, with some extra commands allowed. The complete list of BASIC statements, described in the BBC User Guide, which work with the disc filing systems in the same way as with tape is:

BGET#,BPUT#,CHAIN,CLOSE#,EOF#,INPUT#,LOAD,OPENIN,OPENOUT, PRINT#,SAVE

The BASIC statement OPENUP is not described in earlier User guides, and is not available with BASIC version 1. It has the same syntax as OPENIN, and OPENOUT, but the file is opened for update. OPENOUT performs the same action in BASIC version 1.

The EXT# and PTR# statements are described in the BBC User Guide, but are not available with the tape system. EXT# returns the byte length of an open file. PTR# either returns the pointer of an open file, or resets the pointer to a user specified value. The PTR# statement enables the BASIC programmer to use random access files.

Transferring from tape to disc

If you have had your BBC for some time you will have some programs stored on tape, and will probably prefer to have these stored on disc. If these are commercial programs which use machine code this is not always easy, and we would recommend you use one of the programs sold especially for doing this. If the program which you wish to transfer is written in BASIC and is not too large, then it will transfer to disc quite easily using the following procedure.

Type

```
*TAPE
```

this selects the tape filing system. Now load the program in the usual way, if the program is the next one on the tape you may type

```
LOAD ""
```

press play on the tape recorder, and wait for the program to be loaded. When it has loaded reselect the disc filing system by typing

```
*DISC
```

and save the program to disc. Make sure that the name of the file is only seven characters long. If you wished to call the program "FRED" you would type

```
SAVE "FRED"
```

Acorn DFS and Opus DDOS both require memory for their operation, and lift 'PAGE' from &0E00 to &1900. This reduces the memory available for programs and will prevent those programs which use nearly all available memory, and those which use machine code in fixed positions from working.

Using Text Files

The filing systems provide some simple utilities to manipulate text files. To create a text file called, say, "NIMZO" from the keyboard, type

```
*BUILD NIMZO
```

The filing system will now prompt with a line number for each line of text to be entered into the file :0.$.NIMZO. When you have typed all the lines you want, press the ESCAPE key to terminate the file creation. The screen might now show:

```
>*BUILD NIMZO
   1 Wherein lies the beauty of 3 P-K5...?
   2 Why is this move strong?
   3 The answer is as simple as it is astonishing.
   4 The move is strong because it is weak!
>
```

To check the contents of NIMZO, you can type

```
*LIST NIMZO
```

to display the text in NIMZO with line numbers, or

```
*TYPE NIMZO
```

to display the text without line numbers.

Note that *LIST and *TYPE are ony suitable for text files. Another command *DUMP should be used with non-text files. It displays not only the printed representation of any text in a file, but also the hexadecimal coding of the characters therein.

If you *LIST or *TYPE or *DUMP a long file, it is usually best to turn on the BBC page mode first, by typing CTRL-N. If you want to stop listing before the end of a file, just press ESCAPE at any time.

One common use of text files is to encapsulate an oft-repeated sequence of commands, which can then be executed using the *EXEC command. So for example:

```
>*BUILD START
    1 *TV 0,1
    2 *FX 6,0
    3 *FX 12,3
    4 *DIR 0
>
```

This is a sequence of commands which you may always wish to execute when you switch on the BBC. Now that they are captured in the text file START, you can execute them all by typing

```
*EXEC START
```

and the filing system will behave just as if the four OS commands had been typed in at the keyboard.

There is an even neater method of executing a sequence of commands. First the commands need to be in a text file called $.!BOOT. There is no need to rebuild the file - just change its name:

```
*RENAME START !BOOT
```

Now type the command

```
*OPT 4,3
```

This sets the 'auto-start option' of the current default volume to be EXEC. (For discs in single density format there is only one volume on each surface of the disc.) What this means is that now, whenever you press SHIFT and BREAK keys together, releasing the SHIFT key first, and the default volume is the same as the volume written to above, and the current disc is in drive 0, then the file !BOOT will be EXECed.

Text files can also be used for manipulating BASIC program files. Chapter 37 of the BBC User Guide shows two ways of merging two BASIC programs. The first uses the command *SPOOL, which sends all screen output to a file as well. The other, rather confusingly, uses the *LOAD command, which loads files directly into memory. Normally you should use the BASIC statement LOAD to load BASIC program files into memory.

Ambiguous File Specifications

An ambiguous file specification is the same as a normal, unambiguous, file specification, with two extra possibilities.

The character '#' can be used instead of the directory name, and/or instead of one or more characters in the file name. It represents 'any character'.

The character '*' can be used instead of the directory name, and/or instead of the last characters of the file name. It represents 'any sequence of characters'.

So the ambiguous file specification J### represents any file in the default directory with a 4-character filename starting with 'J', and *.JI* represents any file in the default volume with a filename of 2 or more characters, starting with 'JI'.

Many filing system commands accept ambiguous file specifications, and the interpretation is always 'perform the command action for all the files which match the specification'.

Displaying Catalogues

The *CAT command lists all the entry names in a volume catalogue, together with information about the volume and the disc. It requires a volume to be specified:

```
*CAT
```

will display the catalogue of the default volume, whereas

```
*CAT 2
```

will display the catalogue of the default volume on drive 2.

The *INFO command displays the byte length and the load and execution addresses of all the files matching an ambiguous file specification. Thus:

```
*INFO *.*
```

lists the attributes of all the files in the default volume.

Copying Files

When the file START was renamed to !BOOT above, only the catalogue entry was changed to show the new name. To move the file to another volume or disc, or to make a second copy of it, one of the filing system copying commands must be used.

All the filing system copy commands will replace existing files of the same name, and they all use memory as a buffer while copying files, so you will lose the current program if you have not first SAVEd it.

The *COPY command makes a copy of a file to a different volume (a different drive for single density) with the same entry name. So:

```
*COPY 0 1 f.fred
```

creates a file :1.f.fred on the lower drive from the file :0.f.fred on the upper drive.The command syntax requires a source and target volume, and the entryname of the source file which will be given to the target file.

If the source and target drives are the same, *COPY assumes that the files are on different physical discs, and alternately asks for the source and target discs to be loaded into the drive. This is useful when you have only one drive and want to copy from one disc to another.

A powerful feature of *COPY is that it will copy more than one file at a time, if an ambiguous file specification is given.

```
*COPY 1 0 ##2
```

will copy all files with the 3-character file names ending with '2' in the default directory on drive 1 to the default volume on drive 0.

```
*COPY 0 2 *.*
```

will copy all files from the default volume on drive 0 to the default volume on drive 2.

Earlier the inherent vulnerability of data on floppy discs was pointed out. To protect your valuable programs and data, you should take frequent copies of all your discs, and keep copies well away from the masters and the computer.

The simplest way to do this is to use *BACKUP, which makes an exact copy of a complete disc. Thus

```
*BACKUP 0 1
```

copies the disc in drive 0 to that in drive 1. The target disc must already have been formatted to the same density and number of tracks as the source disc, and any information on it will be overwritten.

With only one drive, you can still use *BACKUP:

```
*BACKUP 0 0
```

directs the user to alternately insert source and target discs, just like *COPY.

Note that *BACKUP uses all available memory as a buffer, and you will lose your current program if you have not previously saved it. If you are performing a *BACKUP on a single drive system, the number of times you will be asked to swap the source and target discs will depend upon the amount of free memory. You are advised to select screen mode 7, which provides the maximum free memory, and hence the least number of disc swaps.

Deleting Files

The converse of copying files is deleting them. The *DELETE command will delete a single file, providing its lock is not on (*ACCESS is used to manipulate file locks).

*WIPE and *DESTROY both accept an ambiguous file specification. *WIPE asks the user if each matching unlocked file is to be deleted, whereas *DESTROY lists all the matches and asks the user if they should all be deleted or not.

5 Commands

The commands described in this section all relate to the manipulation of discs and disc files, but they fall into three categories: operating system commands, filing system commands and utility commands.

Operating system (OS) commands are available on all BBC computers, and under any filing system. However, the interpretation of the command is supported by the active filing system and therefore may vary depending on which filing system is active. Thus the *CAT command is an operating system command which is interpreted somewhat differently by a disc filing system from the tape filing system.

Filing system (FS) commands are supported by the disc filing system, and are specific to the disc operations performed by the disc filing system. They are unlikely to be meaningful, or to be interpreted in the same way, when another filing system is active. Thus *DESTROY is a disc filing system command which is meaningless to the tape filing system.

Utility commands are supported by the disc filing system, but will function correctly whatever filing system is active. Thus *LIST is a disc filing system utility which can be used to list the contents of a disc or a tape text file.

The syntax of the commands is described in a similar style to that used in the BBC User Guide (see p.199). Unlike BASIC, however, the disc filing systems do not distinguish between upper-case and lower-case letters. Thus *DELETE fred and *delete FRED are interpreted as the same.

All commands are of the form:

```
* <command> {<parameter>}
```

<command> is from three to seven characters. Usually a command may be abbreviated to a leading substring followed by a full stop. How short the substring may be will depend on what other commands might share the same abbreviation. Thus *DRIVE can be abbreviated to *DRIV. or *DRI. or *DR., but not to *D. which would be interpreted as *DISC.

For filing system commands there is an additional complication in that any sideways ROMs fitted to the BBC will be given the chance to service the abbreviated command before the filing system. Therefore, the shortest abbreviation given with each command may not be appropriate for your BBC.

Each parameter must be preceded by at least one space, and should not contain embedded spaces except within string contents. Some syntactic items that occur in the parameters of several commands are defined here.

<table>
<tr><td><drv></td><td>drive name</td><td>0|1|2|3</td></tr>
<tr><td><drvsp></td><td>drive spec</td><td>[:] [<drv>]</td></tr>
<tr><td><sch></td><td>spec char</td><td>Any character but #:.*</td></tr>
<tr><td><dir></td><td>directory name</td><td><sch></td></tr>
<tr><td><dirsp></td><td>directory spec</td><td><drvsp> [<dir>]</td></tr>
<tr><td><file></td><td>file name</td><td>{<sch>}x7</td></tr>
<tr><td><entsp></td><td>entry spec</td><td>[<dir>.]<file></td></tr>
<tr><td><fsp></td><td>file spec</td><td><drvsp> <entsp></td></tr>
<tr><td><asch></td><td>ambiguous spec char</td><td><sch> | #</td></tr>
<tr><td><adir></td><td>ambiguous dir. name</td><td><asch> | *</td></tr>
<tr><td><afile></td><td>ambiguous file name</td><td>{<asch>}x7 | {<asch>}x6 *</td></tr>
<tr><td><aentsp></td><td>ambiguous entry spec</td><td>[<adir>.] <afile></td></tr>
<tr><td><afsp></td><td>ambiguous file spec</td><td><drvsp> <aentsp></td></tr>
</table>

When any part of a drive or directory spec is omitted, the corresponding current default is substituted. Thus if the current default drive number is 1, and the current default directory is $, then the file spec JOE specifies the file :1.$.JOE.

*ACCESS <afsp> [L]

Abbreviation: *A.
Type: FS

If the "L" is included, all files matching the ambiguous file spec are locked. If the "L" is omitted, all files matching the ambiguous file spec are unlocked.

A locked file cannot be deleted, overwritten, or written to until the file is unlocked.

The *CAT and *INFO commands display whether a file is locked or not.

Example

```
>*ACCESS A.TEST L
```

locks the file A.TEST

```
>*ACCESS $.*
```

unlocks all the files on the current default drive in directory $.

WARNING
Locking a file does not prevent the files being destroyed by formatting or using the command *BACKUP.

*BACKUP <source drvsp> <target drvsp>

Abbreviation: *BAC.
Type: FS
Related: *COPY, *ENABLE

The disc in the source drive is copied in its entirety to the disc in the target drive, which must be of the same density and length.

If both drive specifications are the same, the user is prompted to insert alternately the source and target discs. Less disc changes will be required if screen mode 7 is used during this command.

This command overwrites the data in user memory, and any information on the target disc. The *ENABLE command must be used before the *BACKUP command is used.

Example

```
*ENABLE
*BACKUP 0 1
```

Copies the disc in drive 0 to the disc in drive 1.

WARNING

The contents of user memory are overwritten by this command, and the original contents of the destination disc are lost.

*BUILD <fsp>

Type: utility
Abbreviation: *BU.
Related: *EXEC, *LIST, *TYPE

The specified file is created directly from the keyboard input. The user is prompted for the input with the current line number. Every character typed is written sequentially to the file. Input is terminated at any time by pressing the ESCAPE key.

*BUILD is usually used to create text files, but it is possible to enter non-printable characters via the keyboard - see the BBC User Guide.

Example

```
>*BUILD !BOOT
1    *KEY10 CHAIN"MENU"|M
2    CHAIN "MENU"
3
>
```

creates a text !BOOT file.

*CAT <drvsp>

Type: OS
Abbreviation: *.
Related: *ACCESS, *DIR, *DRIVE, *FORMAT, *INFO, *LIB, *OPT4, *TITLE

The catalogue of the specified disc is displayed. Also displayed is the disc title, the drive number, the current default directory, and the current default library.

The files are displayed in alphabetic order, with those files in the current default directory being listed first.

Example

```
*CAT 0
GAMES 1 (12)
Drive 0                  Option 2 (EXEC)
Directory :0.$           Library :0.$

    !BOOT L                  BATBALL
    CHASE L                  CRUMYANA

A.  MENU L
```

*COMPACT <drvsp>

Type: FS
Abbreviation: *COM.
Related: *BACKUP

The files on the specified disc are compacted so that all free sectors are in a single block at the end of the volume. This becomes necessary when the free space has become fragmented and there is no single block large enough to create a new file.

Compacting does not destroy any information on the disc. However, if a disc error occurs during compaction, it might be difficult to recover the contents of the file that was being compacted at the time. Therefore it is advisable to make a backup copy of the disc first.

Example

```
>*COMPACT 1
Compacting drive 1
$.!BOOT       L  000000    000000    00001A    002
$.CHASE          001900    FF8023    000370    003
A.MENU        L  001900    FF8023    000130    007
Disk compacted 187 free sectors
```

WARNING

This command overwrites the contents of user memory. If a disc error occurs during compaction the disc is likely to become unreadable. It is wise to make a backup copy of the disc before compaction.

*COPY <source drvsp> <target drvsp> <aentsp>

Type: FS
Abbreviation: *COP.
Related: *BACKUP

All files on the specified source disc matching the ambiguous entry specification are copied, without change of name, to the target disc. If a file of the same name already exists on the target disc, it is overwritten by the copy unless locked.

Example

```
>*COPY 0 1 A.*
```

Copies all files in directory A from the disc in drive 0 to the disc in drive 1.

WARNING

User memory is overwritten.

*DELETE <fsp>

Type: FS
Abbreviation: *DE.
Related: *DESTROY, *WIPE

The specified file is removed from its catalogue, and all the data in the file is lost. Locked files cannot be deleted without first unlocking them using the *ACCESS command.

Example

```
>*DELETE A.MENU
```

Deletes the file MENU in directory A from the disc in the current default drive.

*DESTROY <afsp>

Type: FS
Abbreviation: *DES.
Related: *DELETE, *WIPE

All unlocked files matching the ambiguous file specification are listed, and the user is prompted as to whether they should be destroyed. If so, all the files are removed from the catalogue, and the data in them is lost. The *ENABLE command must be issued immediately prior to this command, or the message 'not enabled' will be issued.

Example

```
>*ENABLE
>*DESTROY :1.A.*
A.MENU
A.FRED
A.BALL

Delete (Y/N) ?Y
Deleted
```

*DIR <dirsp>

Type: FS
Abrrev: *DI.
Related: *CAT, *DRIVE

The default drive and directory are set to those in the directory specification. If any field is not specified, the current default remains unchanged.

The current default values are displayed by the *CAT command.

Example

```
>*DIR S
```

Sets the current directory to S on the current default drive.

```
>*DIR :1.A
```

Sets the current directory to A on drive 1

*DISC | *DISK

TYPE: utility
Abbreviation: *DIS.
Related: *TAPE

The current filing system (e.g. the tape filing system) is suspended, and the disc filing system becomes the active filing system. Any files open under the tape filing system will remain open, and can be accessed again when the tape filing system is reselected using *TAPE.

DRIVE <drvsp>

Type: FS
Abbreviation: *DR.
Related: *CAT, *DIR

The default drive is set to the specified drive number. The current default values are displayed by the *CAT command.

*DUMP <fsp>

Type: utility
Abbreviation: *DU.
Related: *LIST, *TYPE

The contents of the specified file are displayed (dumped) on the screen. Each line of the display is as follows:

```
<addr>  <hex value> <hex value>..     <pvalue> <pvalue>..
```

where

<addr> is the address of the line in the file, in hexadecimal;
<hex value> is the hexadecimal representation of the file character;
<pvalue> is the printed representation of that character, or a full stop if the character is not printable (i.e. has an ASCII value less than 32 or greater than 126 decimal).

The display can be stopped by depressing CTRL and SHIFT together, or page mode may be engaged by typing CTRL N.

*ENABLE

Type: OS
Abbreviation: *EN.
Related: *BACKUP, *DESTROY

Filing system commands which can erase files must be enabled immediately before use. This extra step helps to prevent the accidental erasure of files.

Example

```
>*ENABLE
>*BACKUP 0 1
```

*EXEC <FSP>

Type: OS
Abbreviation: *E.
Related: *BUILD, *LIST, *TYPE

The data in the specified file is read character by character as though those characters were being typed in at the keyboard.

Short command files are usually produced by use of the *BUILD command.

Programs may be merged by *SPOOLing one program, loading a second program, and *EXECing the spooled first program back into memory.

Example

```
>*EXEC PROGTEXT
```

Further input is taken from the ASCII file PROGTEXT until the end of the file is reached.

WARNING

*EXEC should only be used with text files; strange and possibly alarming results may otherwise ensue. BASIC program files are not ASCII files and should not be *EXECed.

*HELP

Type: OS
Abbreviation: *H.

If typed alone, a list of keywords on which further help is available is displayed on the screen.

If *HELP is followed by the name of the disc filing system (e.g. DFS, or DDOS) the syntax specification for all the commands is displayed. If *HELP is followed by UTILS the syntax for the utility commands is given.

*INFO <afsp>

Type: FS
Abbreviation: *I.
Related: *CAT, *LOAD, *RUN, *SAVE

The catalogue details for all files matching the ambiguous file specification are displayed. Each file is displayed as:

```
<ent> [L] <laddr> <eaddr> <byte length> <start sector>
```

where

<ent> is the file's entry name;
L indicates that the file is locked;
<laddr> is the file's load address, in hexadecimal, which will be used by *LOAD and *RUN;
<eaddr> is the file's execution address, in hexadecimal, which will be used by *RUN;
<byte length> is the file's byte length, in hexadecimal;
<start sector> is the address of the file's first data sector of the file, in hexadecimal, relative to the start of the disc.

Example

```
>*INFO BATBALL
$.BATBALL     L   FF1900   FF8023   00023A   00C
```

The file called BATBALL in directory $ has a load address of &1900, an execution address of &8023, a length of &23A bytes, and starts at sector &C (12) on the disc. If it is a single density disc, there are 10 sectors per track, so the file starts at track 1, sector 2.

LIB <dirsp>

Type: FS
Abbreviation: *LIB
Related: *CAT, *RUN

The library drive and directory are set to those in the directory specification. If any field is not specified, the current library value remains unchanged.

When an *command is entered, if the command is not recognised by any of the sideways ROMs (including the disc filing system ROM), then the current default directory will be searched for a file of that name. If a file of that name is not found, then the current library directory will be searched for a file of that name. If no file is found in either the current default directory or the library directory, then a 'file not found' message will be issued. If a file of the correct name is found it will be loaded into memory at its load address, and executed from its execution address. It should be a machine code program.

The *RUN command uses the library directory in the same way.

The current library values are displayed by the *CAT command.

Example

```
>*LIB :1.Z
```

The library is set to directory Z on drive 1.

*LIST <fsp>

Type: utility
Abbreviation: *LI.
Related: *DUMP, *TYPE

The contents of the specified file are displayed as numbered lines of printable characters.

*LIST should be used only with text files, strange and possibly alarming results may otherwise ensue.

BASIC programs are stored in tokenised form, and cannot be sensibly displayed using the *LIST command. An ASCII text file of a BASIC program may be obtained by *SPOOLing a BASIC program.

Example

```
>*LIST !BOOT
1    *KEY 10 CHAIN "MENU"|M
2    CHAIN "MENU"
```

The text file !BOOT on the current default directory, such as that produced by the *BUILD command, is listed.

*LOAD <fsp> [<load address>]

Type: OS
Abbreviation: *L.
Related: *INFO, *RUN, *SAVE

The specified file is loaded directly into memory, at the address specified if given, otherwise at the file's load address, as displayed by *INFO. The address should be specified in hexadecimal without the leading &.

*LOAD is normally used to load binary program and data files. BASIC programs should be loaded using the BASIC statement LOAD.

Example

```
>*INFO FRED
$.FRED      L   FF1900   FF8023   00023C   002
>*LOAD FRED
```

The file FRED has a load address of &1900, as shown by the use of the *INFO command. The *LOAD command will load the file FRED from the current default directory into memory starting at location &1900 as no load address was specified.

```
>*LOAD FRED 2500
```

The file FRED on the current default directory will be loaded into memory from &2500.

*OPT 1 <value>

Type: OS
Abbreviation: *O.
Related: *INFO

This command controls the information displayed every time a file is accessed. If <value> is zero then no information will be displayed when a file is accessed. If <value> is 1 or greater, then the file information (as displayed by the *INFO command) is displayed every time a file is accessed.

Example

```
>*OPT 1 1
```

File information is enabled.

```
>*OPT 1 0
```

File information is disabled.

*OPT 4 <value>

Sets the action which will occur when the disc in the current default drive is present in drive 0, and a SHIFT/BREAK is pressed.

If <value> is 0, then no action will take place on auto-start.
If <value> is 1, then the file $.!BOOT will be *LOADed.
If <value> is 2, then the file $.!BOOT will be *RUN.
If <value> is 3, then the file $.!BOOT will be *EXECed.

To use the auto-start feature, insert the disc in the current default drive and enter the *OPT 4 command to set the desired auto-start option. Place the prepared disc into drive 0, and press SHIFT and BREAK together, release the BREAK key and then release the SHIFT key.

If the auto-start option is active, and no file $.!BOOT is found, the error 'File not found' is reported.

Example

```
>*OPT 4 3
```

Sets the disc in the current default drive to perform an *EXEC of the $.!BOOT file when the disc is placed in drive 0, and SHIFT/BREAK is pressed.

*RENAME <fsp> <entsp>

Type: FS
Abbreviation: *RE.

The directory, and/or the filename of the specified file are changed to those given in the entry specification. The contents of the file are not changed.

The command will fail if a file already exists with the new name, or if the file to be renamed is locked.

Example

```
>*RENAME A.FRED B.TOM
```

Renames the file called FRED in directory A on the disc in the current default drive to TOM in directory B on the same disc.

*RUN <fsp> <parameters>

Type: OS
Abbreviation: *R.
Special abbreviated form: *<fsp> <parameters>

The specified file is loaded into memory at its load address, and then execution of it commences at its excution address. If the file cannot be found in the specified directory, the library directory is searched for a file with the same filename.

The parameters are available to the executing program by using an OSARGS call described in chapter 8.

Use of a file type other than a binary program may cause strange, possibly alarming and feasibly catastrophic results. In particular, BASIC programs should not be *RUN; the BASIC statements CHAIN or LOAD should be used.

Example

```
>*RUN MCODE
```

Causes the machine code file called MCODE, which is in the current default directory or the library directory, to be loaded at its load address, and executed from its execution address. The same effect could have been obtained by typing in *MCODE.

*SAVE <fsp> <start addr> <end addr> [<laddr> [<eaddr>]]
*SAVE <fsp> <start addr> +<length> [<laddr> [<eaddr>]]

Type: OS
Abbreviation: *S.
Related: *LOAD, *RUN

The contents of memory starting at <start addr> and finishing at <end addr>-1, or at <start addr> +<length>-1, are saved as the specified binary file. If a file of that name already exists, and it is not locked, then it is overwritten.

The load and execution address of the file are set to <laddr> and <eaddr> respectively, if specified, or to <start addr> if not. These default addresses are used by *LOAD and *RUN.

All addresses, and <length>, should be specified in hexadecimal without a leading &.

Example

```
>*SAVE MCODE 1900 +2000 2400 2403
```

The contents of memory starting at &1900 of length &2000 is saved to the file MCODE in the current default directory, with a load address of &2400 and an execution address of &2403.

```
>*SAVE MCODE 1900 3900 2400 2403
```

This command performs the same action, using the alternate form of specifying the length.

***SPOOL** [**<fsp>**]

Type: FS
Abbreviation: *SP.

When <fsp> is specified, most subsequent screen output is also directed to the specified file. If a file of that name already exists, and if it is not locked, then it is overwritten.

To avoid infinite loops, output generated by the filing system itself is not sent to the spool file.

*SPOOL without <fsp> specified closes the spool file and terminates spooling.

This command is used to produce ASCII text files of BASIC programs and for copying the output of programs to a text file for subsequent analysis or for incorporation into a document using a word processor.

Example

```
>*SPOOL PROGTXT
>LIST
>*SPOOL
```

The program currently in memory will be saved as an ASCII text file called PROGTXT in the current default directory.

*TITLE <title>

Type: FS
Abbreviation: *TI.
Related: *CAT

The title of the disc in the current drive is set to the string <title>. The title of a disc is displayed by *CAT

The title may comprise a maximum of 12 characters. If less are specified, the string is made up with spaces. If the string is to contain embedded spaces, it should be enclosed by a pair of double quotation marks.

Example

```
>*TITLE "GAMES DISC"
```

The title of the disc in the current drive is changed to 'GAMES DISC'.

*TYPE <fsp>

Type: utility
Abbreviation: *TY.
Related: *DUMP, *LIST

The contents of the specified file are displayed as lines of printable characters.

*TYPE should be used only with text files; strange and possibly alarming results may otherwise ensue.

This command is similar to the *LIST command, but line numbers are not displayed.

Example

```
>*TYPE :1.A.TXT2
```

Displays the contents of the ASCII text file called TXT2 in directory A on the disc in drive 1.

***WIPE <afsp>**

Type: FS
Abbreviation: *W.
Related: *DELETE, *DESTROY

For each unlocked file matching the ambiguous file specification, the user is prompted as to whether the file should be deleted; if so, the entry is removed from the catalogue and its contents are lost.

Example

```
*WIPE #.A*
```

All unlocked files on the disc in the current drive in any directory, which start with A will be listed individually. The user will be prompted to confirm the deletion of each file. Those files for which deletion is confirmed will be removed from the catalogue.

6 DDOS Extended Features and Commands

DDOS provides a number of features to help overcome the limitations inherent in Acorn DFS, while maintaining a high degree of compatibility.

The floppy disc controller fitted with DDOS is capable of operating in both single and double density. DDOS allows the use of both types of disc; single density discs are identical to discs produced on an Acorn DFS system, double density discs provide 80 per cent greater capacity, and up to 248 files.

DDOS increases the number of files which may be held on one side of a disc by dividing the disc into up to eight volumes. Each volume is equivalent to a surface of a disc of the Acorn format single density, with the same limit of 31 files, and the same catalogue information.

Volumes may be explicitely referenced by including a volume letter immediately after a drive number in a file specification for example

```
:2D.F.JAZZ
```

specifies the file called JAZZ in directory F of volume D on drive 2. The default volume can be set by using the *DRIVE command.

DDOS maintains its current default volume through BREAK, allowing use of the auto start feature from any volume on any disc.

Extra commands are provided to control the extra features of DDOS and provide more information to the user.

Extended Syntax for DDOS

<vol>	volume name	A|B|C|D|E|F|G|H
<volsp>	volume specification	<drvsp> [<vol>]
<drvol>	drive-volume	[:[<drv>] [<vol>].]
<dirsp>	directory specification	<drvol> [<dir>]
<fsp>	file specification	<drvol> <entsp>
<afsp>	ambiguous file spec	<drvol> <aentsp>

Additional Commands available IN DDOS and CHALLENGER.

*4080 <setting>

Type: FS

Sets the filing system so that 40-track discs may be used with an 80-track drive.

If setting is ON then only 40-track discs may be used in an 80 track drive. A disc error will be produced if the command is issued when 40-track disc drives are used.

If setting is OFF then only 40-track discs may be used in a 40-track disc drive, and only 80-track disc drives may be used in an 80-track disc drives.

If setting is AUTO then the filing system will allow both 40-track and 80-track discs to be used in an 80-track disc drive.

The default is OFF.

Example

```
>*4080 AUTO
```

Sets the computer so both 40-track and 80-track discs may be used in 80-track drives.

*CAT <volsp>

When the *CAT command is used with DDOS or CHALLENGER it is assumed that the command is folowed by a volume specification. If no volume letter is explicitly included, then the default volume, A, on the specified disc will be displayed.

Example

```
>*CAT 0B
```

The catalogue of volume B on drive 0 will be displayed.

```
>*CAT 1
```

The catalogue of the default volume on the disc in drive 1 will be displayed, this is volume A.

*COMPACT <volsp>

A volume specification is expected following the *COMPACT command. If no explicit volume letter is used, then the default volume will be compacted.

*DENSITY <arguement>

Type: FS
Abbreviation: *DEN.
Related: *CAT

DDOS by default automatically detects the density of the discs which are used, making it simple to use both single and double density discs. The protection system used by some commercial discs prevents the proper operation of this automatic density detection.

This command allows the disc interface to be configured so that either only single density discs are readable, or only double density discs are readable.

*DENSITY AUTO is the default condition. When this condition is set, single and double density discs may be used, and their density is automatically detected.

*DENSITY SINGLE sets the controller to assume that the disc to be read or written is a single density disc. This allows some protected discs still to be used with DDOS.

*DENSITY DOUBLE sets the controller to assume that the disc to be read or written is a double density disc.

After power up or a CTRL-BREAK, DENSITY is set to AUTO, a BREAK or SHIFT-BREAK does not change the existing setting of DENSITY.

Example

```
>*DENSITY SINGLE
```

Only single density discs will now be correctly read and written.

*DRIVE <volspec>

A volume specification is expected following the *DRIVE command. This allows the default volume to be set to a volume other than volume A.

*FDCSTAT

TYPE: FS

Returns the contents of the status register of the floppy disc controller. If you discover an error in the DDOS operating system, enter *FDCSTAT, and note the values returned. Include these values when reporting the error to Opus.

*FORMAT

Type: FS
Related: *VOLGEN

New floppy discs are usually in an unformatted form. This means that they cannot be used to store information until after they have been formatted. The utility *FORMAT will prepare a disc in either single or double density as required. A disc formatted in single density may be read by any Acorn

DFS compatible disc system. An Acorn DFS system will be unable to read a disc formatted in double density.

The format program is invoked by typing

```
*FORMAT
```

You will be asked to enter the number of tracks per side (40 or 80), the drive in wish you wish to format the disc, and the density in which you wish the disc to be formatted. You will be given a last chance to change your mind before formatting (formatting wipes all information from a disc!). If an error occurs while formatting a disc you will be given a message 'verify error'.

*MAP <volspec>

Type: FS
Abbreviation: *MA.
Related: *CAT, *STAT, *COMPACT

Displays a map of the specified volume, showing the tracks and sectors occupied by files on a volume, and the tracks and sectors which are free space.

Example

```
>*MAP 0C
```

Displays sectors occupied by files and free space on volume C of the disc in drive 0.

*MCOPY <source volsp> <target volsp>

Type: FS
Abbrev: *MC.
Related: *COPY, *BACKUP

All files on the source volume are copied to the target volume, overwriting the previous contents of the target volume. The command makes maximum use of the memory of the BBC computer to maximise the transfer speed, and this command is to be preferred when making backup copies of the files on a disc.

Example

```
*MCOPY 0A 1C
```

All files in volume A on drive 0 are copied to volume C on drive 1, overwriting the previous contents.

WARNING

This command overwrites user memory, and the previous contents of the target volume.

*SROM <value>

Type: FS
Abbreviation: *SR.
Related: *SAVE, *LOAD

Specifies the socket number (0 - F) of the sideways ROM or RAM which is to be read by the *SAVE command when saving data in the address range &8000 to &BFFF, or the sideways RAM which is written to by a *LOAD to the address range &8000 to &BFFF.

Example

```
>*SROM F
```

Selects socket &F, (15), such that a subsequent *SAVE to disc of the memory area &8000 to &BFFF will save the data in the ROM in socket 15.

*STAT <volsp>

Type: FS
Abbrev: *ST.
Related: *CAT, *XCAT

The size of the volume and the remaining space in the volume is displayed in sectors and Kbytes.

Example

*TAPEDISK <fsp>

Type: FS
Abbrev: *TAPED.

The tape filing system is used to search for the file specified. When found it is transferred to the current default directory using the same filename. If the filename is greater than 7 characters, the file name is truncated to 7 characters.

Example

```
>*TAPEDISK TAPEFILE
```

The file called TAPEFILE will be searched for on tape and when found transferred to the current default directory, given the name TAPEFIL.

*VERIFY <drvspec>

Type: FS
Related: *FORMAT

*VERIFY allows you to check a disc to ee if it has become corrupted. Under normal use this will not occur unless the discs are misused.

Example

```
*VERIFY 0
```

Verifies the disc in drive 0.

*VOLGEN [<drvspec>]

Type: FS
Abbreviation: *VOLG.

Related: *FORMAT, *XCAT, *CAT

Volumes are automatically generated at format time. For double density discs it is possible to change the amount of space on the disc allocated to each volume. Single density discs have only one volume with the total disc space allocated to it. To change the default size of the volumes on a double density disc use the *VOLGEN command.

*VOLGEN reads the configuration sector from the floppy disc and produces a display which shows volumes A to H and their allocated sizes in Kbytes.

To deallocate space from a volume, type the volume name followed by RETURN. To allocate space to a volume, type the volume letter followed by the space in Kbytes you wish to allocate.

Disc space is always allocated to volumes in full tracks, so the space allocated will be rounded down to the nearest full track.

Note that using *VOLGEN wipes existing information from a disc, and so must be *ENABLEd before use.

Example

```
>*ENABLE
>*VOLGEN
```

***XCAT <drvspec>**

Type: FS
Abbrev: *X.
Related: *CAT

The catalogues of all the volumes on the specified drive are displayed. If no files exist in a volume, no catalogue header will be displayed.

7 CHALLENGER Extensions

CHALLENGER provides either 256Kbytes or 512Kbytes of additional memory, organised as one or two additional 'disc drives' to the BBC Microcomputer. These additional disc drives are accessed as drives 4 and 5. The memory is arranged as pages of 256 bytes on the JIM bus (&FD00 - &FDFF), using adresses &FCFF and &FCFE on the FRED bus as paging registers.

The additional RAM discs are treated as single density discs having one volume, capable of storing 31 files. By dynamically allocating sectors to files, the 'can't extend' error is eliminated.

The drive numbers associated with all the physical drives may be reconfigured using the new *CONFIG command, so that a RAM disc may be set up as the logical drive 0. This allows the RAM disc to be used with software which insists on accessing drive 0 for its data.

CHALLENGER makes use of the memory residing on its board for the workspace required by the disc filing system, allowing PAGE to be left at &0E00. This provides a greater compatibility with tape based programs.

Extended Commands in CHALLENGER

*CAT

This command is extended to allow the * wildcard to be used instead of the volume letter. When this is used, the catalogue of all the volumes on the selected disc are displayed.

Example

```
>*CAT 1*
```

The catalogues of all the volumes on drive 1 are displayed.

*CONFIG <log drvsp>=<phys drvsp> <log drvsp>=<phys drvsp> ..

Configures the physical drive specified to be referenced by commands using the logical drive specification.

Example

```
>*CONFIG 0=4 4=0
>SAVE ":0.A.FRED"
```

Sets the first RAM disc (physical drive 4) to be logical drive 0, and physical drive 0 to be logical drive 4. The current program is saved as a file called FRED in directory A on the RAM disc currently configured as drive 0.

*OPT 6 <value>

Controls the density assumed by the filing system when reading discs. The default setting 0 selects automatic recognition mode, whereby the filing system will correctly identify the density of all but very few protected discs without user intervention. A value of 10 forces the filing system to consider all discs to be single density, a value of 18 forces all discs to be considered as double density. Other values are inappropriate with standard format discs.

Example

```
*OPT 6 10
```

All discs will be considered to be single density, a disc error will be produced if an attempt is made to read from a double density disc.

*OPT 7 <value>

Immediately sets the step rate used with the floppy disc drives. The step rate at power up is controlled by links 3 and 4 on the keyboard printed circuit board. This may be changed using *FX 255 and resetting using the BREAK key. This command allows the stepping rate to be changed without resetting the computer.

if <value> is 0 then the step rate is 30 ms;
if <value> is 1 then the step rate is 20 ms;
if <value> is 2 then the step rate is 12 ms;
if <value> is 3 then the step rate is 6 ms.

*OPT 8 <value>

Controls the 40/80 track mechanism. The default value 0 turns off this mechanism; the filing system expects 40-track discs in a 40-track drive and 80-track discs in a 80-track drive. A value of 255 turns on the automatic mode, whereby both 40 and 80 track discs may be read in an 80 track drive. This mode is slightly slower than others, and is not appropriate for some protected discs. A value 1 selects 40-80 mode which effectively converts an 80-track drive into a 40 track-drive.

Example

```
*OPT 8 255
```

Both 40 and 80-track discs may be read in a 80-track drive.

*OPT 9 <value>

Specifies the socket number (0-15) of the sideways ROM which is to be read by the *SAVE command when saving data in the address range &8000 to &BFFF.

*VOLGEN [<drvspec>]

This command performs the same action as in DDOS, however the size of a volume is specified in sectors, and all current volume space allocation is displayed in sectors and in Kbytes.

When space is requested for a volume, providing sufficient space is available on the disc, the requested space is allocated. Since the space on a track cannot be assigned between two volumes, any remaining space on a track at the end of a volume will remain unallocated.

This means that the maximum volume size which may be allocated is now &3FF sectors.

8. Assembler Interface

The disc filing systems all support the standard OS file system vectors. These are:

OSFILE	Manipulate a whole file, or its catalogue entry.
OSARGS	Manipulate an open file's pointer/miscellaneous.
OSBGET	Read one byte from an open file.
OSBPUT	Write one byte to an open file.
OSGBPB	Read or write bytes from an open file/miscellaneous.
OSFIND	Open or close files
OSWORD	(with A = &7F) Read or write sectors direct to disc.

In the ensuing descriptions, unless otherwise stated:

(i) all addresses are low byte first;
(ii) all 32-bit addresses can refer to the I/O or second processor memory; 8-bit or 16-bit addresses refer to the I/O processor memory only;
(iii) all registers are preserved through calls;
(iv) the status flags Z,N,C,V are not defined on exit.

OSFILE Call address &FFDD Indirected via &0212

Load or save a whole file.

On entry: A = action;
X,Y point to parameter block.

Parameter block:
&00-01 point to file spec, terminated by &0D
&02-05 Load address D.
&06-09 Execution address E.
&0A-0D Length L (start address S if A=0)
&0E-11 Lock K (finish address F if A=0)

A=0 Save the block of memory from D to D+L-1 as the specified file,replacing any existing file of the same name (cf *SAVE).
A=1 Change the addresses and lock of the specified file to D,E and K.
A=2 Change the load address of the specified file to D.
A=3 Change the execution address of the specified file to E.
A=4 Change the lock of the specified file to K (the lock will be set on if either of bits 1 or 3 are set in byte &0E).

A=5 Set D,E,L and K to the values for the specified file (cf *INFO).
A=6 Delete the specified file (cf *DELETE).
A=&FFLoad the specified file into memory, either at D (if &06 = 0), or at the file's load address (cf *LOAD).

On exit A = 1.

OSARGS Call address &FFDA Indirected via &0214

Load or save the data about a file.

On entry:
A = action;
X points to 4-byte zero page param P (if marked *);
Y = 0 or open channel number.

A=0, Y=0	Set A = 4, indicating disc filing system
A=1*, Y=0	Set P to the address of the parameters of the last *RUN command or equivalent.
A=&FF,Y=0	Flush all file buffers from memory.
A=&FF,Y=chan	Flush the specified buffer.
A=0*, Y=chan,	Set P to the file pointer (cf BASIC PTR#)
A=1*, Y=chan,	Set the file pointer to P (cf BASIC PTR#)
A=2*, Y=chan,	Set P to the file length (cf BASIC EXT#)

OSBGET Call address &FFD7 Indirected via &0216

Gets one byte from a file.

On entry: Y = open channel number.

On exit:
If file pointer < file length then
C = 0;
A = data byte indexed by the pointer;
pointer = pointer + 1
else
C = 1;
A undefined.

OSBPUT Call address &FFD4 Indirected via &0218

Writes one byte to a file.

On entry:
Y = open channel number;
A = value to be written to the data byte indexed by the file pointer.

On exit:
pointer = pointer + 1;
if pointer > length then length = pointer.

OSGBPB

Call address &FFD1 Indirected via &021A

Read or write a byte or group of bytes to or from a specified file.

On entry:
A = action;
X,Y point to parameter block.

Parameter block: (for A = 0 to 4)

&00	open channel number.
&01 - 04	data address D.
&05 - 08	number of bytes L.
&09 - 0C	file pointer P.

A=1 Set the file pointer to P, then as for A = 2.
A=2 Write L bytes from D to the specified file.
A=3 Read L bytes to D from the specified file.

On exit:
(L′ bytes actually read or written);
File pointer = pointer + L′;
D = D+L′;
L = L - L′;
if L′ < L then C = 1 else C = 0.

Parameter block: (for A = 5 to 7)

&00	not used.
&01 - 04	data address D.

A=5 Set D to the sequence: binary 12; the 12-character title of the default volume; its binary auto-boot option.
A=6 Set D to the sequence: binary 1; the single character default drive number; binary 1; the single character default directory.
A=7 Set D to the sequence: binary 1; the single character library drive number; binary 1; the single character library directory.

On exit:
C = 0.

Parameter Block: (for A = 8)

&00	access count U.
&01 - 04	data address D.
&05 - 08	number of filenames L.
&09 - 0C	catalogue pointer P.

A=8 Set D to a sequence of the first (P = 0) or next L filenames from the default directory. Each filename is represented as binary 7 followed by 7-characters.
Set U to the default volume's access count.

On exit:
(L' filenames actually listed);
P = P + L'*8;
D = D + L'*8;
L = L - L';
if L' < L then C = 1 else C = 0.

OSFIND **Call address &FFCE Indirected via &021C**

Open or close a file.

On entry: (close)
A = 0;
Y = 0 or open channel number.

Y=0 Close all open files.
Y=chan Close specified channel.

On entry: (open)
A = open mode;
XY point to file specification, terminated by &0D.

A=&40 Open specified file for input.
A=&80 Open specified file for output.
A=&C0 Open specified file for update.

On exit: If open succeeded then A = channel number else A = 0.

OSWORD **Call address &FFF1 Indirected via &020C**

OSWORD &7F allows commands to be sent to the floppy disc controller. The commands are those required by the Intel 8271. Double density disc filing systems all must use an alternative floppy disc controller in order to support double density operation. The commands entered into OSWORD

&7F are translated into the commands required by the double density floppy disc controller by the disc filing system.

On entry:	A = &7F; XY point to parameter block, this may be situated in either I/O or second processor memory.

Parameter block:

&00	drive number
&01-04	32 bit data address
&05	number of parameters n (3 for read or write sector)
&06	fdc command
&07	first parameter (track T.)
&08	second parameter (sector S)
&09	third parameter (length L)
..	additional parameters
&07+n	error flag E.

The following commands are supported by DDOS and CHALLENGER.

INITIALISE
SEEK
READ DRIVE STATUS
READ/WRITE SPECIAL REGISTERS
READ I.D.
READ DATA
WRITE DATA
VERIFY DATA
READ DATA AND DELETED DATA
WRITE DELETED DATA
FORMAT TRACK

INITIALISE

Initialise the step rate of the floppy disc controller.

0	Drive		
1-4	Not used		
5	2 (not used by DDOS)		
6	command	35	(00110101)
7	parameter	0D	(00001101)
8	step rate	(see below)	
9	not used		

Step rate: only the two least significant bits of the parameter are used.

11 (3)	6 ms.

10 (2)	12 ms.
01 (1)	20 ms.
00 (0)	30 ms.

SEEK

0	Drive		
1-4	Not used		
5	1 (not used by DDOS)		
6	command	29	(XX101001)
7	Track address	0-255	

READ DRIVE STATUS

0	DRIVE		
1-4	Not used		
5	0 (Not used by DOS)		
6	Command	2C	(XX101100)
7	Result		

Result Bit

7	0	
6	RDY1	always 1
5	0	
4	0	
3	WR. PROT.	=1 if write protected
2	RDY0	always 1
1	0	
0	0	

READ/WRITE SPECIAL REGISTERS

0	Drive
1-4	Not used
5	1 (Not used by DDOS)
6	Command
7	Register address
8	Data if write, result if read

Command

Read	3D	(XX111101)
Write	3A	(XX111010)

A READ ID. command should be issued prior to reading these registers in order for them to be updated by the filing system.

DDOS registers

Register 0	40 track disc in 80 track drive flag

Bit 7	Used by DDOS
Bit 6	0 = normal, 1 = perform double track stepping
Bits 5-0	always 0

Register 3	Density control flag

Bit 7	used by DDOS
Bit 6	0 = single density, 1 = double density
Bits 5-0	always 0

Register 4	Sideways ROM select

For Read/Write operations in sideways ROM/RAM area (address &8000 to &BFFF) this register determines which ROM socket is selected during read/write operations. It takes on a value 0-&FF.

Register 12	Track address

Holds the track number at which the floppy disc controller thinks its at.

READ ID

Read the ID of the first sector found on the track.

0	Drive
1-4	address for ID bytes C,H,R,N
5	3 (Not used by DDOS)
6	Command
7	Track number
8	&40 (not used by DDOS)
9	&41 (not used by DDOS)
10	Result

The READ ID. command will write the first 'sector header' information (C,H,R,N) to the address given by bytes 1,2,3,4. These bytes are described as follows:

C	The cylinder number (track number) of the track at which the drive is positioned. The cylinder number of a track at format time does not necessarily correspond to the physical track. Tracks may be formatted with track numbers up to 255, a device often used in disc protection schemes.

H	The Head parameter injected at format time. By definition, this should be 0 for side 0 and 1 for side 1. Many writers of formatting programs have ignored this parameter. This parameter may take any value from 0 to 255, and thus may be employed for disc protection.
R	The record number of the sector found. Usually the same as the sector number. Record numbers are usually 0 to 9 for single density, and 0 to 17 for double density. The value can be in the range 0 to 255 and thus may be employed for disc protection.
N	The size of the sector, normally 1 but may take the values 0,1,2,3,4.

N=0	128 byte sector
N=1	256 byte sector
N=2	512 byte sector
N=3	1024 byte sector
N=4	2048 byte sector

The READ ID command in DDOS will try both single density and double density for the correct ID. Two attempts will be made in each density before an error is returned.

READ DATA

Read up to the maximum number of sectors on a track.

0	Drive		
1-4	Address to put data		
5	3 (Not used by DDOS)		
6	Command	13	(XX010011)
7	Track number		
8	Start sector		
9	Sector length/number of sectors		(lllnnnnn)
10	Result		

Sector length/number of sectors
lllnnnnn where lll indicates the sector length (see READ ID)
nnnnn indicates the number of sectors

WRITE DATA

0	Drive		
1-4	Address from which to write data		
5	3 (Not used by DDOS)		
6	Command	0B	(XX001011)
7	Track number		
8	Start sector		

9	Sector length/number of sectors		
10	Result		

VERIFY DATA

0	Drive		
1-4	Not used		
5	Not used		
6	Command	1F	(XX011111)
7	Track number		
8	Start sector		
9	Sector length/number of sectors		
1-	Result		

READ DATA AND DELETED DATA

0	Drive		
1-4	Address to put data		
5	3 (Not used by DDOS)		
6	Command	17	(XX010111)
7	Track number		
8	Start sector		
9	Sector length/number of sectors		
10	Result		

WRITE DELETED DATA

Write sectors with a deleted data address mark.

0	Drive		
1-4	Address from which to write data		
5	3 (Not used by DDOS)		
6	Command	0F	(XX001111)
7	Track number		
8	Start sector		
9	Sector length/number of sectors		
10	Result		

9 Disc formats.

Single density format

The disc is divided into 40 or 80 tracks, numbered from 0 (innermost) to 39 or 79 (outermost). Each track consists of 10 sectors, numbered 0 to 9. Each sector holds 256 data bytes.

The first two sectors of track 0 contain the single volume catalogue; all other sectors are available for files. The format of the catalogue is as follows:

Sector 0

&00-07		First 8 characters of title.
&08-0F		First entry:
	&08-0E	File name;
	&0F	directory; bit 7 set if locked;
&10-17		Second entry.

Sector 1

&00-03		Last 4 characters of title.
&04		Access count, in packed decimal
&05		Number of entries in catalogue * 8.
&06		00bb00nn, where bb = auto-start boot option; nn = high order 2 bits (of 10) of number of sectors on the disc.
&07		Low order 8 bits (of 10) of number of sectors on the disc.
&08-0F		First entry:
	&08	Low order 8 bits (of 18) of load address;
	&09	Middle order 8 bits (of 18) of load address;
	&0A	Low order 8 bits (of 18) of execution address;
	&0B	Middle order 8 bits (of 18) of execution address;
	&0C	Low order 8 bits (of 18) of byte length;
	&0D	Middle order 8 bits (of 18) of byte length;

	&0E	eellddss, where
		ee = high order 2 bits (of 18) of execution address;
		ll = high order 2 bits (of 18) of length;
		dd = high order 2 bits (of 18) of load address;
		ss = high order 2 bits (of 18) of start sector;

	&0F	Low order 8 bits of start sector.
&10-17		Second entry.

...

The 18 bit addresses are interpreted as follows: if the two high order bits are both set, the address refers to the I/O processor. Otherwise the address refers to the second processor.

Double Density format

The disc is divided into 40 or 80 tracks, numbered from 0 (innermost) to 39 or 79 (outermost). Each track consists of 18 sectors, numbered 0 to 17. Each sector holds 256 data bytes.

Track 0 contains volume A catalogue in sectors 0 and 1, volume B catalogue in sectors 2 and 3, and so on up to volume H catalogue in sectors 14 and 15. There is a disc catalogue in sector 16; sector 17 is not used. All other tracks are available for files. The format of the disc catalogue is as follows:

Sector 16

&00	Configuration/version number	
&01-02	Number of sectors on the disc.	
&03	Number of sectors per track.	
&04	Number of tracks on the disc - 1	
&05-07	Not used.	
&08-09	Volume A entry:	
&08		First track, or 0 if not allocated;
	&09	Not used.
&0A-0B	Volume B entry.	

The first sector of each volume catalogue allocated is the same as sector 0 of the single density volume catalogue.

The second sector is similar to sector 1 of the single density catalogue, except the number of sectors on the disc is replaced by the number of sectors in the volume, and the start sector is relative to the first sector of the volume.

10. Error Messages.

&BF Can't extend

An attempt has been made to increase the length of a file beyond its allocated sector length, but another file, or the end of volume follows immediately on the disc.

&C0 Too many files

Only five files can be open at one time. An attempt has been made to open a sixth file.

&C1 File read only

An attempt has been made to write to file opened for read only.

&C2 File open

An attempt has been made to open a file already open, or an operation such as *DUMP, *SAVE, or *INFO has been carried out on an open file.

&C3 File locked

An attempt has been made to write to, delete, or rename a file which is locked.

&C4 File exists

An attempt has been made to rename a file, but the new name is the same as that of an existing file.

&C5 Drive fault NN at TT/SS

This error implies that the disc drive is faulty, and may need repair. NN is the floppy disc controller error listed below, TT and SS are the track and sector at which the error occured.

&C6 Disc full

An attempt to open a file for output, or to save a file, has failed because there is insufficient total space left on the disc. *COMPACT will often free more space, but will overwrite a program in memory. It is wise to keep a spare, unformatted disc available for use in emergencies.

&C7 Disc fault NN at TT/SS

An attempt to access the disc at the track and sector specified by TT and SS has failed. NN is the floppy disc controller error listed below.

Disc Faults	
08	clock error clock bit missing during a read opration, suspect the disc controller.
0A	Late DMA The processor did not respond fast enough, suspect the disc controller.
0C	ID field CRC error The cyclic redundancy code error check on the sector id field failed; this may be caused by a faulty disc, a badly inserted disc, or a faulty drive.
0E	Data field CRC error The cyclic redundancy code error check on the data field failed; this may be caused by a faulty disc, a badly inserted disc, or a faulty drive.
10	Drive not ready The disc stopped rotating at the required speed, usually caused by a badly inserted disc, or a badly warped disc.
14	Track 0 not found The controller could not find track 0. This fault is usually generated by an attempt to read or write data to an unformatted disc.
18	Sector not found The controller could not find a sector. This is often caused by an unformatted disc, a badly corrupted disc, or a disc of the wrong type, eg a 80 track disc in an 40-track drive.

&C8 Disc changed

An attempted access of an open file has failed because the volume or disc on which it was opened has been changed.

&C9 Disc read only

An attempt to write to a disc has failed because the disc has a write protect tag fixed to it.

&CB Bad option

An *OPT has been used with an invalid parameter.

&CC Bad filename

A filename which is too long, or contains illegal characters has been used.

&CD Bad drive

A drive number not in the range 0 to 3 (for Acorn DFS and DDOS) or in the range 0 to 5 (for Challenger) has been specified.

&CE Bad directory

A directory of more than one character, or an illegal character has been specified.

&CF Bad attribute

The *ACCESS command has been issued with an attribute other than L following it.

&D6 File not found

The file specified has not been found in the specified directory.

&DC Syntax

A command has been recognised by the disc filing system, but the parameters have been missed or are incorrect.

&DE Channel

An attempt has been made to access a file on a channel which is not open.

&DF EOF

EOF stands for end of file. An attempt has been made to read beyond the end of a data file.

&FE Bad command

An * command has been issued which has not been recognised by the computer, and no file of the same name has been found on the current default directory, or the library directory.

11. DDOS System Memory Usage

DDOS requires 11 pages of memory for its use as a filing system. Nine pages are for general usage and can be claimed by other software packages. Two pages are regarded as private to DDOS.

The 9 pages of static memory are used by DDOS as follows:

Page E **&0E00 - &0EFF**

Each and every time a file is opened this page is loaded with the first sector of the disc catalogue. In double density, however, this page is first used to load the configuration sector from track 0 sector 16 which provides the track offsets to the volumes allocated. The appropriate catalogue is then loaded.

Page F **&0F00 - &0FFF**

Each time a file is opened this page is loaded with the second sector of the disc catalogue.

Pages 10 and 11

&1000 - &10DF

General DDOS usage such as Flags, Disc ID tables, Pointers, save registers, filename buffers,OSGPB control blocks, OSFILE control blocks, TUBE flags, file channel parameters, counters, default parameters, selected parameters.

&10E0 - &10E5

'Disc specific' parameters such as Density, Sectors/track for current disc input/output operations.

The following memory is used for disc parameters associated with open files.

&10E6 - 10EB	Disc parameters for open file 1
&10EC - 10F1	Disc parameters for open file 2
&10F2 - 10F7	Disc parameters for open file 3
&10F8 - 10FD	Disc parameters for open file 4
&10FE - 1103	Disc parameters for open file 5

The following memory is used for disc parameters for the source and target drives for COPY, MCOPY, and BACKUP commands.

&1104 - 1109	Disc parameters for source drive
&110A - 110F	Disc parameters for target drive
&1110 - 111F	Spare

The following memory is used as channelsector buffers of 256 bytes each. One buffer is used for each of the five possible open files.

&1200 - 12FF	Sector buffer for file 1
&1300 - 13FF	Sector buffer for file 2
&1400 - 14FF	Sector buffer for file 3
&1500 - 15FF	Sector buffer for file 4
&1600 - 16FF	Sector buffer for file 5

Private Workspace

The following two pages are regarded as 'private' to DDOS and by their very nature should not be used by any other program. This, however, is often abused, and so its usage is described as follows:

Page &17

When a different filing system is selected (eg Teletext), this page is used to save important information which would otherwise be overwritten in the static workspace by the variables of the new filing system. Such information would be file control blocks and other parameters associated with open files.

Page &18

This page is used by the *BUILD command to buffer a line input from the keyboard.

Page &19..

Page &19 and onwards is free for normal usage. The value of PAGE as used in BASIC will take on the value &1900.

Note

If ROM software in addition to DDOS and BASIC is resident, then the 'private workspace' may be moved to pages higher in memory as the additional ROMs place demand for additional static workspace, and demand their own private workspace.

12. DDOS Installation

The disc upgrade can only be fitted to a BBC Microcomputer Model B, fitted with the 1.2 or later version of the operating system. If you are not sure which operating system you have fitted, type

```
*FX0
```

If the 1.2 operating system is fitted the computer will print out

```
OS1.2
```

If you have an earlier version of the operating system contact your dealer to have your machine upgraded.

The disc upgrade kit contains the following items.

2 off	7437 or 7438	(IC 79,80)
1 off	14 pin header	(IC86) link pin 1 - pin 4
1 off	16 pin header	(IC87) link pin 9 - pin 12
1 off	Controller board	
1 off	27128	(DDOS EPROM)

IC81 and IC82 are not required with DDOS.

Do not remove the devices from their packing at this stage. Some of these devices are fabricated using MOS technology and can be damaged by static electricity.

Installation Procedure

1. Switch off the computer and remove the power plug from the mains supply.

2. Remove the two screws at the rear of the machine, and the two screws on the bottom of the machine which are nearest the front edge of the machine. In earlier models these screws are marked 'FIX'. Carefully lift off the cover of the machine.

3. Undo the two or three nuts and bolts which hold the keyboard to the lower part of the case. Disconnect the ribbon cable which connects the keyboard to the motherboard. Disconnect the lead which goes to the loudspeaker. If speech is fitted, disconnect the second ribbon cable.

4. Lift the keyboard clear of the computer, and lay it down carefully.

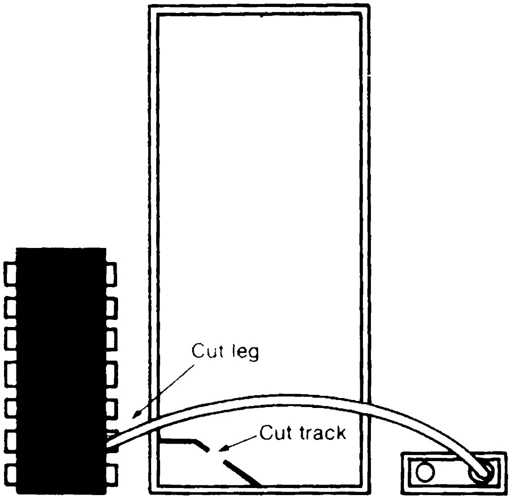
5. Printed on the main circuit board will be the issue number. If the issue number is 1,2 or 3 a circuit board modification is required. **DO NOT MODIFY ISSUE 4 OR LATER BOARDS WITH THE EXCEPTION OF CUTTING THE WIRE LINK S9 IF IT IS PRESENT.**

6. Modification for boards of Issue 1,2 and 3.

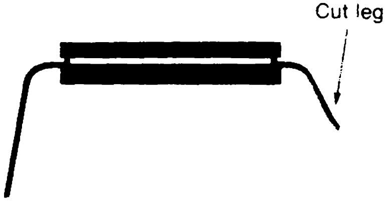
(a) Identify pin 9 of IC27. Cut the leg with a fine pair of side cutters as close to the printed circuit board as possible. Bend up the leg.

(b) Cut the the track between pin 9 of IC27 and the link S9 using a fine knife or scalpel. Do not cut any other tracks.

(c) Solder a fine wire between the leg of pin 9 of IC27 and the East pad of S9 (as viewed from the front of the computer).



PIN 9 of IC27 going to right hand block of S9.



(d) On Issue 1 and 2 boards solder a wire link across link S8.

7. If there is a wire link across S9, cut it, if no wire is apparent then the link is not made, and no change is necessary. Set the following links if they are not already made.

S18	North
S19	East
S20	North
S21	two links, both East/West
S22	North
S32	West
S33	West

8. Carefully insert ICs 79, 80, 86, 87 in the marked sockets, ensuring that the notch of each IC is to the North. (The same as the ICs already in the board.) To insert some of the ICs into the sockets it will probably be necessary to carefully bend the rows of pins in slightly, this is most easily done against a hard flat surface. If these ICs are already fitted then leave them in the board.

   Insert the headers in IC86 and IC87 taking care to get the correct orientation. If your computer already has chips installed in locations C79, 80, 86, and 87, then leave them as they are, and modify the DDOS controller board as follows. Locate pin 18 of the WD1770 floppy disc controller on the controller board. Carefully cut the printed circuit track leading from pin 18. Solder a fine wire to pin 18 of the WD1770 and solder the other end carefully to pin 1 of IC86 on the BBC motherboard.

9 Insert the DDOS ROM in position IC101, to the East of the circuit board, this ensures that no other ROM containing programs called FORMAT, or VERIFY are accessed instead of DDOS. If an extension ROM board is used ensure that the DDOS ROM is put into the socket with the highest priority (normally 15).

10 Check all the ICs for bent legs and correct orientation, if all looks correct refit the speaker wire (the polarity of this does not matter) and the ribbon cable connecting the keyboard to the main circuit board, folding it back under the keybard so that it will be out of the way of the disc controller board. On some computers this is easier if the cable is removed and swapped round, however the cable must not be left with a twist in it. If spare cable protudes from the connectors, fold this over the connector so that it lies flat.

11 Carefully insert the controller board into position IC78. Be sure all 40 pins line up correctly and the base seats evenly into the socket on the main mother board. The board does not have to be forced into place! The bulk of the board sits over to the West of the socket.

12 Plug in the computer, and switch on. If all is well the following message sould be displayed.

```
BBC Computer 32K

OPUS DDOS (version number)
BASIC
>
```

If this or a similar message is not obtained, switch off the computer, remove the mains plug, and check all the steps you have taken. Also check the position of the ribbon cable which connects the keyboard to the main circuit board.

13 Remove the mains, and refit the bolts holding the keyboard in position. Refit the lid of the computer, locating the LEDs in their holes in front of the keyboard. Replace the screws at the back and underneath the computer. Replace the mains plug, and turn on for a final check.

13 CHALLENGER Installation

The CHALLENGER disc SYSTEM can only be fitted to a BBC Microcomputer Model B, fitted with the 1.2 or later version of the operating system. If you are not sure which operating system you have fitted, type

```
*FX0
```

If the 1.2 operating system is fitted the computer will print out

```
OS1.2
```

If you have an earlier version of the operating system contact your dealer to have your machine upgraded.

The Challenger system contains the following items.

1 Challenger disc unit
2 DDOS EPROM
3 Header

Installation Procedure

1 Switch off the computer and remove the power plug from the mains supply.

2 Remove the two screws at the rear of the machine, and the two screws on the bottom of the machine which are nearest the front edge of the machine. In earlier models these screws are marked 'FIX'. Carefully lift off the cover of the machine.

3 Undo the two or three nuts and bolts which hold the keyboard to the lower part of the case and move it aside to reveal the five 28-pin IC sockets situated in the bottom-right area of the main circuit board (looking from the front).

4 Insert the EPROM in the right hand socket, labelled IC101. If there is already a ROM in that socket, move it down to the left. It is essential that the ROM in the fifth socket from the right is not disturbed. When inserting the EPROM it will probably be found that the legs need to be bent in slightly, this is best done against a firm surface.

5 Locate S9 (next to the Acorn Logo) and ensure that there is a wire link connecting the two connection points of S9 (in accordance with Acorn specification). If this is not the case then check to see whether there is a chip in IC78. If there is then no further action is required. If there is neither a chip in IC78 nor a link across S9, then you must insert the 40-pin socket supplied into the IC78 socket. Ensure the polarity is correct, the inserted socket should link pins 11 and 20 of the IC78 socket.

6 Reposition the keyboard, and plug in the computer to check that it initialises itself. It should give the normal sign on message, but say 'CHALLENGER not connected'. This is because the main Challenger unit has not yet been connected. If you do not get a similar message, switch off the computer, and check your earlier steps. If you cannot see any fault, contact your local dealer or Opus for assistance.

7 Turn off the computer and remove the mains plug. Re-install the keyboard, and the lid, taking care to locate the LEDs correctly in their holes.

8 Connect the six pin power lead to the auxiliary power socket under the computer, and insert the ribbon cable connector into the connector marked 1MHz Bus. DO NOT CONNECT IT TO THE DISK DRIVE CONNECTOR. The red edge of the ribbon cable indicates pin 1 on the connector, and goes to the right hand side of the connector, marked with an arrow on the case of the computer.

9 Reconnect the power to the computer and switch on. The computer should provide its usual sign on message, and announce either

```
Opus Challenger 256
```

or

```
Opus Challenger 512
```

or a similar message. You are now ready to use your Opus Challenger Disc System.

GUARANTEE

This Opus product is guaranteed for a period of 2 years from the date of purchase, subject to the conditions listed below:-

1 The guarantee shall only be effective if the whole guarantee form is completed and returned to Opus Supplies Limited within ten days of the products purchase.

2 The guarantee applies only to the original purchaser and is not transferable. The warranty applies only to products purchased from an authorised dealer to whom the equipment was sold by the company or direct from Opus Supplies.

3 All repairs and modifications to the unit must be undertaken by the company via the dealer network. The warranty is immediately invalidated if this procedure is not followed. Where possible claims under the guarantee should be referred to the dealer from where the unit was purchased. If this is not possible you should contact Opus direct. When making a claim the customer's portion of the certified guarantee must be produced.

4 Customers are liable for the carriage costs of units returned under the claims procedure.

5 If, after testing by the company, no fault can be found with the system and the unit complies with the published specification, the company reserves the right to charge a reasonable fee for testing and returning the unit.

6 The liability of the company under this guarantee is limited to the cost of repair or complete replacement (at the discretion of the Company) of the defective equipment.

7 The Company does not accept responsibility for any loss or damage during transit to or from the dealer or the company, but in this event every effort will be made to investigate complaints of loss or damage if they should arise.

8 This guarantee does not affect the purchaser's statutory rights.

GUARANTEE REGISTRATION

MODEL/INVOICE/REF.NO.

DATE OF PURCHASE

DEALER..

ADDRESS...

- -

GUARANTEE REGISTRATION

OPUS SUPPLIES LIMITED COPY

MODEL/INVOICE/REF.NO

DATE OF PURCHASE

DEALER..

ADDRESS...

I accept the terms of this guarantee

Customer's signiture...

Print name...

This guarantee is only valid if it is completed and signed above by the customer and returned within ten days of the purchase by the original user to Opus Supplies Ltd., 55 Ormside Way, Holmethorpe Industrial Estate, Redhill, Surrey. RH1 2LW.

This guarantee is valid in the U.K. only.

Opus.
Opus Supplies Ltd.,
55 Ormside Way
Holmethorpe Industrial Est.
Redhill, Surrey
Tel. No. Redhill 65080