

# CO-PROCESSOR FILING SYSTEM

=====

for BBC Master 512  
or BBC Model B/B+ with 80186 Second Processor

## INTRODUCTION

=====

The Essential Software Co-Processor Filing System (CPFS) allows the memory of a Master 512 co-processor to be used as a ramdisc (or memory disc) by the BBC micro when operating in native mode. The 512 may be fitted to any model of 8 bit BBC micro, that is, a Model B, B+ or Master. In the case of the Master, the 512 can be fitted to either the internal or external Tube. CPFS may be used with any of these setups.

The amount of file storage provided by CPFS depends only on the memory size of the 512. CPFS self-configures to use all of the available 512 RAM except for a small fixed overhead of approximately 20K for the Tube transfer software and of course for the directory space needed to store file information. Using CPFS you can load and save files in commercial applications, in your own programs, or manually, all using normal and familiar BBC Micro filing system commands.

However, there's one major difference - CPFS operates very much more rapidly than any floppy disc. Loading and saving of whole files takes place at over 100K per second, or put another way, loading a 10K program takes as long as the blink of an eye. Data transfers using the MOS call "OSGBPB" are equally rapid. Owing to operating system

overheads simple "BPUT"s and "BGET"s are not speeded up quite so much, but even so are still considerably faster than with floppy discs.

## HOW IT'S DONE

=====

Users of Model B or B+ machines will know that when the BBC micro is already running and they then switch on the 512's co-processor adaptor, nothing happens until BREAK or CTRL-BREAK is pressed. Within limits the BBC micro can continue to operate as if the 512 were still sitsje off. In effect the micro does not know that the 512 has been powered-up until a reset is performed.

Similarly the Master 128 can be used in BBC mode while the 512 is disabled using the MOS command '\*CONFIGURE NOTUBE'. In this case the micro also behaves as if it had no co-processor, even though the memory used by the 80186 co-processor (the 512) is actually present and powered up.

In both the above cases CPFS can take control and modify the behaviour of the complete system to make the 512's memory available for file storage while the BBC micro continues to be used in native mode.

Do remember that all the data stored in any memory disc is lost as soon as either machine is turned off. Also, of course, when the second processor is used for its original purpose the same memory is used as CPFS uses for files, so any data stored in the memory disc will be corrupted if the Tube is turned on. If you use CPFS to store anything that you are going to want to use again, make absolutely sure that it is written onto a "proper" disc before switching off or enabling the Tube. The program provides a method of doing this easily.

Because of the substantial differences in both the hardware and software of the machines, CPFS operates slightly differently in a Model B or B+ from the way it operates in the Master. However, in practice these differences are few, most of them are transparent to the user and none of them affect the facilities or capabilities provided by CPFS.

In the following instructions all points apply to any type of installation unless specific notes to the contrary are given.

"Model B" means either a Model B or B+ micro with a 80186 second processor attached to the external Tube using a co-processor adaptor, while "Master" means a Master 128 with a 512 fitted either internally to the main circuit board, or externally using a co-pro. adaptor.

All trademarks and tradenames referred to in these instructions are acknowledged as the property of their respective owners.

## INSTALLATION

=====

CPFS is contained in a 16K EPROM which must be plugged into a vacant ROM socket in the BBC micro. For technical reasons CPFS cannot be run in sideways RAM.

Before installation of the EPROM you must ensure that the micro, the 512, the monitor and any other connected devices are switched OFF. Note that if the computer has recently been in use the monitor screen may retain a considerable static electric charge which you might pick up without realising it. Such a charge can damage EPROMs and some of the chips inside your micro.

Before taking the lid off your micro or handling the CPFS (or any) EPROM, ensure that you are not carrying a static charge. If you're in any doubt touch a water tap or a radiator to discharge yourself before starting the job and don't wear (e.g.) a nylon shirt in a dry air environment. **THIS IS NOT A JOKE!**

Take care to ensure that the EPROM is inserted the correct way round. You will notice a small notch at one end of the chip. When viewed from the operator's position, i.e. in front of the keyboard, this notch must point to the rear of the micro in a Model B or B+, to the left in a Master. Installing the EPROM reversed will very probably render the chip useless if you switch the micro on. When you are inserting the EPROM take care that all the legs are properly engaged in the socket before applying firm, even pressure to push it home.

In a Master the CPFS EPROM may be either fitted internally, or if you prefer, in a cartridge for external fitting. The resulting effective ROM number is not significant. Note that CPFS does not work reliably if it is used in one of the Quad-ROM cartridges produced by some manufacturers.

For users of Model B or B+ micros CPFS must be fitted in a socket of higher priority (number) than the ROM containing the Tube control code. In practice this means that CPFS must be in a higher numbered socket than other filing system ROMs. If this cannot be done as your machine is currently set-up you will need to 'shuffle' the order of existing ROMs so that this condition can be met. Failure to do so will prevent CPFS from taking proper control of the Tube when required and will prevent it from operating correctly.

When the EPROM is correctly fitted and the machine is powered up, the command:

**\*HELP CPFS**

(or \*H.CP.) displays a summary of CPFS commands. This can be used to check that the EPROM is properly installed.

## GETTING STARTED

=====

In use CPFS operates as an independent filing system, much like any other BBC micro random-access filing system such as DFS or ADFS. It is available whenever the 512 co-processor is powered up but is configured "Off".

If your machine is a Master, when configured "NOTUBE" (using the standard MOS \*CONFIGURE facilities) CPFS is available immediately the machine is switched on.

If the machine is a Master configured as "TUBE", or it is a Model B/B+, the machine will start in "80186 mode" by default and you must de-select the Tube (in software) before using CPFS. Of course if you do wish the 512 to start up, simply proceed as normal. If you wish instead to use CPFS then press BREAK to get the machine's "\*" prompt. (You might need to have an ADFS-compatible disc in Drive 0, though on a Master you may sometimes be able to get away with pressing ESCAPE to stop the disc drive spinning.) Then enter:-

### NOTUBE

followed by RETURN followed by BREAK again. Note that CPFS is programmed to check for the presence of the 512 only at power-up or when BREAK is pressed. This means that if you are using an external co-processor adaptor and have switched the main computer on first, then just like the MOS, CPFS will not know that the second processor is available until you have followed the above procedure.

On configuring NOTUBE and pressing BREAK your 'normal' BBC micro title screen will appear. At this stage CPFS is ready for use. When CPFS is active files can be loaded from or saved to the 53?3Y etc. to swap between the various filing systems. This means that in most applications and situations you can use CPFS just like another disc by selecting it whenever you wish to.

In common with filing system standards CPFS will be selected as the current filing system in one of two ways. (There are more, but these are the usual ones). The first is by entering:

## \*CPFS

(which can be abbreviated to \*CP.) followed by RETURN. The second way is by holding down the "C" key while you press BREAK.

On a Model B or B+, in situations where CPFS is available it will be selected as the current filing system on pressing BREAK. This is because you have inserted it in a higher priority ROM slot than other filing system ROMs. (If this does not happen on a Model B/B+, then you have put the ROM in the wrong place and CPFS will not work correctly. See the installation instructions above.) In these circumstances if you want to boot up a disc you will have to press SHIFT/D/BREAK (for a DFS disc) or SHIFT/A/BREAK (for an ADFS one).

In the Master you can also use CPFS as a temporary filing system in the usual way when DFS or ADFS is the current filing system, by including the name (-CPFS-) in file-names and \*-commands.

When CPFS has been active, to re-enable the Tube to again permit the use of the 512 as a DOS processor, simply enter:

## \*TUBE

and press BREAK (or CTRL/BREAK to start DOS-Plus, of course). Take care that you mean to do this - all CPFS files will be lost immediately.

CPFS will not work as a filing system using any other co-processor than a 512, though other co-processors will work normally with CPFS installed. If you have a Master with two co-processors connected (one internal and one external) and one of these is a 512 then CPFS will locate it and use it. (The INTUBE/EXTUBE configuration setting is ignored.) Note however, that it is NOT possible to run three processors at once. In other words, you cannot use one co-processor, e.g. a 6502, as your language processor while using the 512 as a memory disc.

## FILES AND FILE NAMES

=====

In a CPFS catalogue, files appear very like they do in Acorn DFS, though CPFS is less limited than DFS and even provides extra

facilities over ADFS. The main points about how files are stored in CPFS are:-

The memory disc can hold up to 127 files, provided there is sufficient space.

File-names may be up to 10 characters long, and can contain any characters except one of:

\* # : . " -

Files can be grouped into directories, but as in DFS, directory names are single characters which appear as part of the filename. There is no sub-directory structure, though in a temporary storage medium like a memory disc this is hardly a restriction. You can therefore specify a directory as part of a file-name in the standard Acorn manner by using "." as a separator, thus:-

A.THISFILE

specifies a file called "THISFILE" in directory "A".

Any directory can be selected as the current CPFS directory as usual by the command '\*DIR <directory-letter>'. All files are assumed to be in the current directory if no explicit directory is given in commands. In the usual way the default (root) directory is called "\$".

There is no distinction between upper and lower case letters in filenames, so two filenames such as:

X.Myfile and x.MYFile

will be treated as identical names in all CPFS operations.

Some commands can act on a number of files simultaneously using wildcards and in this CPFS follows the same conventions as the Acorn DFS. For such commands, # represents any single character, \* can represent any group of characters up to the maximum length of a filename.

Up to 5 CPFS files can be open at once. Files can be opened using Basic's OPENIN, OPENOUT and OPENUP commands, or using the MOS call "OSFIND".

When you use floppy discs you often find that a file cannot be extended because there is another one following it on the disc. Alternatively, you may find that you cannot save a new file because there is no single block of free space large enough for it, although there may well be enough unused space on the disc in total. These problems cannot occur in CPFS, because it re-organises free space in the memory disc automatically when it needs to. This means that there is no equivalent of the DFS's "Can't extend" message, nor of ADFS's "Compaction required."

There is no limit (other than the total limit of 127 files) to the number of files that can be in one directory. It is worth noting that a CPFS directory can thus contain more files than directories in other filing systems. (A DFS disc (one side) can normally contain only 31 files in total. An ADFS directory has a maximum size of 47 files.) There are applications (Inter-Word, for example) which sometimes display the contents of a directory. These programs may not be able to handle more than 47 files at once. If you are using an application like this with CPFS then it is up to you to make sure that there are no more than 47 files in any single CPFS directory.

Files in the CPFS memory disc remain intact (assuming that you don't change them yourself) until the machine (either the BBC or the 512) is switched off, or until the Tube is enabled for normal 512 operation. Pressing BREAK or CTRL-BREAK (provided that you have not re-enabled the Tube by \*TUBE) has no effect on the contents of the files or the CPFS catalogue except in one particular circumstance. That circumstance is if you were to press BREAK twice in rapid succession. If you do so then there is a small chance that one or two bytes in the 512 may be corrupted.

When CPFS is not active pressing BREAK always corrupts a few bytes of co-processor memory, but for 512 operation this is of no consequence. CPFS is written to overcome this fact when the 512 is used as a ramdisc, but because of the time it takes the BBC micro to perform a software reset after BREAK has been pressed this protection operation may not succeed if it does not have time to be completed between one BREAK and the next. Since pressing BREAK twice in quick succession serves no useful purpose anyway, remember always to leave at least half a second between successive presses of the BREAK key when CPFS is in use and files are stored in it.

## CPFS COMMANDS

=====

All the filing system commands recognised by the MOS (i.e. the common commands) are available in CPFS, as usual. These are described in detail in your Reference Manual or User Guide and perform the same functions in CPFS as they do in your existing disc filing systems. In CPFS you can therefore use \*LOAD, \*CAT, \*SPOOL etc. just as you have always done.

In addition, the following commands can be used. First the utility commands are shown. These are commands that can be used whether or not CPFS is selected as the current filing system (or as a temporary filing system on the Master). After them are the filing system commands, to use which CPFS must be selected. In each section they are given in alphabetic order.

#### \*CPFS

-----

This command selects CPFS, making it the current filing system. The command can be abbreviated to "\*CP." It will only work if a 512 is connected and was switched on at the last power-up or BREAK. Also the Tube must be de-selected (see \*NOTUBE below) to prevent normal 512 operation or this command will give the error "Bad Tube configuration".

#### \*NOTUBE

-----

This command is used to switch off the Tube to allow CPFS to use the 512 as a file storage device. The minimum abbreviation is "\*NO.". To use it when the co-processor is active, first press BREAK to get the Monitor's "\*" prompt (or use STAR from DOS-Plus, but this will only waste time) then enter:

#### NOTUBE

followed by RETURN. You must then press BREAK again, when the Tube will be off and CPFS will be available on demand. In the Master, this command is exactly equivalent to \*CONFIGURE NOTUBE, including the fact that the setting is preserved while the machine is switched off.

#### \*TRANSFER

-----

This transfers files from one filing system to another and may be

abbreviated to `*TR.`. It is similar to the Master's `*MOVE` command, but is quite a bit faster and considerably more flexible. It can be used to copy a file or a number of files from a directory in one filing system to a given directory on another filing system. In addition, it can be used to copy all the files in the CPFS memory disc (in all directories) onto a physical ADFS or DFS disc.

`*TRANSFER` requires 5 parameters, but because it would be easy to make mistakes if you were obliged to enter them all in a single command line, you are prompted for each parameter as it's required. To use the command first enter:

`*TRANSFER` or `*TR.`

followed by RETURN.

You are first requested to specify the source filing system. Enter "C", "A", "D" or "N" (followed by RETURN) for CPFS, ADFS, DFS or NET respectively. You will not be able to enter more than one letter. (See below under `*OPT7` if you are using a network.)

Note that if ADFS is not the current filing system but you specify it as the source filing system, then you must have an ADFS disc in the ADFS current drive (or Drive 0 if ADFS has not been used since the last BREAK). The reason for this is that `*TRANSFER` causes the specified filing system to initialize and ADFS can only do so if there is a disc present. The transfer does not have to be from this disc, and it does not have to remain in the drive. It is only to let ADFS get started.

You are then asked for the source directory name, when you must enter a name which is valid for the specified filing system. Alternatively, if the currently selected directory is to be the source directory then you may just press RETURN and this will be used. Using DFS, if there is no current directory and you do not specify one then the directory "\$" is used by default. Using ADFS you must supply a directory name if the disc is not already `*MOUNTed`; and on the other hand, if you do supply a directory for ADFS, then a disc with this directory on must be present. If you are using a disc filing system and you want to specify a directory that is not on the current disc, then include the drive specification with the directory. For example, to transfer from the default directory of DFS drive 3 the entry would be:-

:3.\$

Next you are prompted in turn for the destination filing system and then for the destination directory. These are entered in exactly the

same way and with the same rules as for the source filing system and directory, including the default of pressing RETURN to indicate the current directory, and the need for a disc to be in the drive if ADFS is used.

Finally you must enter the file specification for the object(s) which are to be transferred. This may be a single file name fully specified, or an ambiguous file specification using the wild cards # and \*. If, rather than giving a file or file-list specification you just press RETURN, that will be taken to mean that all the files in the given directory are to be transferred and is the same as supplying a \*. The transfer then begins (assuming that the specified file or files exist in the source) and the file names are displayed as each is processed.

If you are transferring files from CPFS to ADFS or from CPFS to DFS (and only in these two situations) there is an extra facility. You can also enter a wild card (\* or #) as the source directory specification. When you do this all files which match the source file specification in all CPFS directories will be transferred. It therefore follows that if, in addition, you have also specified "\*" (or pressed RETURN) when prompted for the file specification, \*TRANSFER will copy every file in the CPFS memory disc onto an ADFS or DFS disc.

If you use this to copy files from CPFS to ADFS, then files in CPFS directory "\$" will be copied into the specified ADFS destination directory (NOTE: not necessarily the root). CPFS files in other directories will be copied into single-character sub-directories of the specified destination directory. These sub-directories will be created automatically if they do not already exist.

Although as you'll see below problems have been catered for in \*TRANSFER, it is sensible whenever possible to ensure that the destination disc and directory have sufficient free entries and free space when transferring files from CPFS to a floppy disc. Remember that CPFS can store up to 127 files and that a 'standard memory 512' ramdisc is larger than any DFS disc (side) while a 1 megabyte 512 provides more storage than either DFS or ADFS floppy discs.

## Errors during \*TRANSFER

-----

Sometimes it may be impossible for \*TRANSFER to copy a file. This might be for a number of reasons such as, for example, that the file already exists in the destination directory and is locked, because the disc becomes full, or because it has a write protect tab.

In these cases the destination filing system (or, much less commonly, the source filing system) will generate an error. When this happens in \*TRANSFER an error message is displayed and you will be asked whether you want to 'Abandon, Continue or Re-try'. Enter only A, C or R in response; there is no need to press RETURN after the entry.

"Abandon" will end the transfer immediately and is the option you should use when the error is permanent or requires immediate investigation, such as 'Drive fault', 'Bad FS map' and so on.

"Continue" will continue the transfer from the next file, omitting the one that caused the error. This is useful, for example, when you have used a wildcard source file specification for convenience, perhaps knowing that one or two of the files already exist and are locked on the destination disc.

"Re-try" will attempt to continue the transfer from the file that caused the error. Of course, in this case if nothing were changed then the same error would probably recur immediately, so first you are asked for a new destination directory.

This gives you the opportunity to insert a new destination disc (in the case of a "Disc full" error, for example) and to continue the transfer. To do so, after the error enter R for Re-try. Next insert your new disc. If you are using DFS, then simply continue, entering the new destination directory name.

However, using ADFS you must be more careful with directory names. Make sure you give the full destination directory name, including the drive specification again. Thus, for example, if you want to continue on a new disc transferring files to the root directory, then insert the new disc in Drive 0 (say) and enter the destination directory name:

:0.\$

By including the drive specification you force ADFS to \*MOUNT the new disc, without which you would probably get a "Disc changed" error.

Note that if you are copying a large file an error message such as "Disc full" might occur when a file is partly copied. In that case you will (already) have an incomplete file on the original destination disc. It is wise to check for this situation after the transfer has finished and immediately delete any incomplete files if this has occurred. Remember that the filename displayed immediately before the error report is the one which will be incomplete.

\*TRANSFER is quite a complicated routine, and there are a number of additional points that are worth noting about its use.

a) \*TRANSFER uses main memory between PAGE and HIMEM, just like disc systems' \*COPY, \*BACKUP etc. This means that any program, word-processor text, spreadsheet etc. which is in memory at the time of the command will be corrupted, regardless of which is the currently selected filing system. Note that this happens immediately you have entered the \*TRANSFER command. Always ensure that any valuable data in memory is saved before using \*TRANSFER.

b) If ADFS is either the source or the destination filing system, any directory name you enter will be taken as the pathname from the current directory. In other words if you are in directory 'FRED' and you supply this name as the destination directory \*TRANSFER will expect to find a sub-directory of that name in the current directory, giving an error if there isn't one. Give the full pathname (including "\$") if you are not sure what the current directory is, and include the drive specification if you want to use a different drive than the current one.

c) ADFS files with the "E" attribute set ("execute-only" files) will not be copied by \*TRANSFER.

d) Under DFS filenames may have a name of a maximum of 7 characters in length. In consequence longer filenames, when transferred from other filing systems, such as CPFS and ADFS, must be truncated.

Sometimes this truncation results in duplicate filenames. For example, if you have two files in CPFS called "NEWFILE1" and "NEWFILE2" and transfer them to DFS, both names become "NEWFILE" when truncated. The result of this is that the first file will be

overwritten by the second on the DFS disc because both truncated names are identical. Of course the CPFS copy of the files will still exist, but since no warning will be given when this happens it is up to you to be aware of and to allow for this circumstance.

- e) The destination filing system must be different from the source.  
\*TRANSFER cannot be used to copy files from one directory to another within the same filing system.
- f) \*TRANSFER will copy files between a DFS disc and an ADFS one, but the discs must be in different drives; there are no pauses to change discs. Because of ADFS' "need for a disc", it will usually be most convenient to have the ADFS disc in Drive 0, and the DFS disc in Drive 1 & 3 (unless Drive 1 is the current ADFS drive).
- g) You can use \*TRANSFER with filing systems other than CPFS, ADFS, DFS and NET, though it will not necessarily work, and you will certainly not be able to use the "wild directory" facility. If you have another filing system in your machine, then you can specify it but \*TRANSFER will not recognize its name. Instead you must give its filing system identity number.
- h) \*TRANSFER will not work with the Tape or ROM filing systems. Neither can it be used to transfer files to or from DOS.
- i) \*TRANSFER selects the specified directories as the current directory in both the source and the destination filing systems. These will remain selected afterwards. The filing system which was current when \*TRANSFER was entered will remain the current one after it has finished.
- j) The speed of \*TRANSFER depends almost entirely on the speed of the filing systems involved. CPFS itself is almost instantaneous, but other systems take more time. Some filing systems are quicker when reading than writing, so for some combinations of filing systems you will find that \*TRANSFER copies much faster in one direction than the other. Transferring a file will be quicker if the file can be copied as a single unit rather than having to be broken up. So for greatest speed use Mode 7 or shadow screen.

\*TUBE

-----

This command re-enables the Tube after it has been disabled with \*NOTUBE and can be shortened to "\*TU.". In the Master it has exactly the same effect as \*CONFIGURE TUBE. In the Model B it works differently, but behaves similarly. In either machine you must follow it by pressing BREAK. Enabling the Tube after using CPFS will, of course, lose all CPFS files immediately.

## Filing System Commands

-----

These commands require CPFS to be selected, either as the current filing system or, in the Master, as a temporary filing system.

### \*ACCESS

-----

This works exactly as in DFS. There is only one attribute - a file can be locked or not. It is used in the form

\*ACCESS <file spec.> L

to lock files, and

\*ACCESS <file spec.>

to unlock them. In this command <file spec.> can be a single file name or an ambiguous filename using wild cards. Locked files cannot be changed in any way unless they are first unlocked. Since CPFS can only be used for temporary storage (i.e. files are lost when the machine is switched off) this command is probably less useful than in permanent storage systems such as DFS or ADFS.

### \*COMPACT

-----

Unlike DFS and ADFS, CPFS automatically re-organises its free space whenever it needs to, so strictly speaking this command is unnecessary. However, the process of compaction is not instantaneous, and can take nearly fifteen seconds (though it is usually much less). This could be a problem if it were to happen in the middle of some time-critical operation, so \*COMPACT can be used to force compaction at any time. The command has two forms:

\*COMPACT

(without parameters) will move all files to the bottom of co-processor

memory, leaving all the free space at the top. The command can also be used with a file name.

`*COMPACT <file-name>`

which will re-organise the free space so that it is all immediately above the named file (which must exist). This means that you can ensure that the named file can expand as much as it needs to without further compaction (provided, of course, that nothing else is later saved in that space or another re-organisation of space forced by a different file extending). Note that, unlike the same command in some other systems, `*COMPACT` in CPFS does not use main memory and so will not corrupt any program or data you have there.

`*DESTROY`

-----

This allows you to delete a number of files with a single command. The syntax is:

`*DESTROY <file spec.>`

where `<file spec.>` may contain wild cards. You are first asked to confirm that you really mean to destroy the files. All files which match the given file specification and which are not locked or open are then listed, and you are asked to confirm once again that you wish them to be deleted. The effects of `*DESTROY` can be drastic, and this is why you are asked whether you want to go ahead both before and after the file-names are displayed. Since a lot of files may appear in the list, the machine is put into "paged-mode" before listing them.

`*DIR`

----

This sets the current directory. The syntax is simply:

`*DIR <dir>`

where `<dir>` is a single character, and can be any except one of:-

`# * : . " or -.`

Note that there is no `'` after the `DIR` command as it is not an abbreviation.

## \*FREE

-----

This shows the amount of free space available. It is displayed in both hex and decimal, showing the amount of free space in the co-processor memory and the amount that is already used, together with the number of files in the catalogue.

## \*LIB

----

\*LIB is used to set the library directory. Just as with DFS and ADFS, if you try to \*RUN a file which does not exist in the current directory, CPFS looks in the library directory for it and runs it from there automatically if it is found.

## \*RENAME

-----

This enables a new name to be given to a file. It takes the form:

\*RENAME <old file-name> <new file-name>

\*RENAME will only rename one file at a time and cannot be used with wild cards for obvious reasons.

## \*WIPE

-----

\*WIPE is used for selectively deleting files. It operates in a similar way to \*WIPE in DFS. Enter:

\*WIPE <file spec.>

and each file that matches the file specification is listed in turn, one at a time. For each one which is not locked or open you are asked whether you want the file to be deleted. Locked and open files are listed along with the other files, but you are not given the option of deleting them.

## Other Filing System Commands

-----

Certain filing system commands are handled by the MOS on the Master, or by the existing filing systems (DFS in fact) in the Model B or B+. The three commands BUILD, LIST and DUMP are available in CPFS but, as in ADFS, are not provided by CPFS itself and are included here for completeness.

## \*BUILD

-----

This command allows lines typed manually at the keyboard to be entered directly into a file. It works just as it does on disc systems. The syntax is:

```
*BUILD <file-name>
```

Each line is entered, followed by RETURN, followed by ESCAPE to close the file after the last line.

## \*DELETE

-----

This deletes a single file, hence wildcards are not permitted. The command takes the form:

```
*DELETE <file-name>
```

## \*DUMP

-----

\*DUMP is used to give a hex and character dump of a file. Enter:

```
*DUMP <file-name>
```

and the contents of the file will be displayed.

## \*INFO

-----

This command is used to give filing system information about a file or a group of files. The format of the command is:

```
*INFO <file spec.>
```

where <file spec.> can include wild cards. Information about all files that match the given file specification is displayed. In 20 or 40 column screen modes the display from \*INFO causes line wrap-round and it is not very useful. However, unlike Acorn's DFS or ADFS, in 80 column modes CPFS includes a heading line to remind you which item is the load address, the execution address, the length and the file location. There might be a lot of files, so the screen is set to paged mode.

For each file, first the file name is shown, followed by L if the file is locked and/or O if the file is open, followed by four numbers, all in hex. These are:

- a) The file's load address
- b) The execution address
- c) The size of the file in bytes
- d) The position of the start of the file in co-processor memory.

Bear in mind that, unlike physical discs, in CPFS there are no sectors or tracks. Therefore the position of each file is shown as the absolute address in the 512 where the file is stored. Because of the memory used by the Tube operating system and the CPFS catalogue, the first available byte in co-processor memory and the lowest possible address for the 'first' file is &5000.

#### \*LIST

-----

This displays the contents of a file in ASCII format line by line, including line numbers, and converting all non-printable characters into control sequences (See \*MTYPE). It is used in the form:

```
*LIST <file-name>
```

#### \*MTYPE

-----

Operates exactly as \*TYPE does in the Master, and provides Model B and B+ users with that extra facility, though the original \*TYPE can still be used as well.

In the B and B+ \*TYPEing a non ASCII file to the screen outputs all characters directly through the MOS routine "OSASCI". The result is that when non-display character codes are encountered the effects are entirely unpredictable, e.g. the screen mode may change, the printer may be enabled, etc. etc. In short the usual result is often effectively a hang up, but it's always a screw-up and never produces a sensible display.

\*MTYPE outputs only valid alphanumeric characters plus line-feeds and carriage-returns directly. All other characters are translated to the extended display format used by GSREAD. That is - the character | is used to denote the start of an extended character. Characters with values below 32 are displayed as | followed by the display character

which would be produced by increasing the non-display character code by 64. For characters above 128 the code minus 128 is used, with |! denoting that the top bit was set. Character 127 ("Delete") appears as |? and character 124 (normally |) as ||. This gives the following displays:-

CHR\$(0) is shown as |@

CHR\$(1) " |A

CHR\$(2) " |B

CHR\$(128) " ||@

CHR\$(129) " ||A

\*PRINT

-----

This command is included with CPFS for the convenience of users of the Model B/B+ since it already exists on the Master. It also displays the contents of a file in ASCII format, except that there are no line numbers and all characters (whether or not they lie within the usual printable range) are sent directly to the VDU driver. This can give strange effects with a non-ASCII file, but it does enable the direct display of graphics. Note that even with a "pure ASCII" file the display will not be the same as when using \*TYPE. This is because, using \*PRINT, character 13 causes a Return but no Line-feed, and so the cursor goes back to the start of the same line.

Notes on Common Commands

-----

Most "common" filing system commands (i.e. those handled by the MOS) work in CPFS exactly as you would expect. The following, though, are worth noting.

\*CAT

----

This displays a list of all the files on the memory disc, with the following additional information:

If a file is locked (with \*ACCESS) then an "L" appears after the file name. If a file is open at that time an "O" is also shown.

Files in the current directory are listed first, followed by the files in other directories, shown in alphabetic order. Each directory begins on a new line.

At the end of the file list the total number of files is displayed, the total number of bytes used by them and the total remaining free space is given. (This information is not given if you are in Screen Mode 2 or 5, because of the difficulty of reading it in these 20-column modes.)

### \*INFO and \*EX

-----

These are common commands on the Master. On the Model B, \*INFO is provided as part of the filing system, while \*EX applies only to ADFS. Both are used to display similar information about files. The note on \*INFO under "Other Commands" (above) describes the format of the information.

### \*OPT

----

This works exactly as in the DFS or ADFS for values of 4 or less, with an extra option necessary for network users.

\*OPT 1,n enables or disables file information display during load and save operations, where any value of n except zero enables the information.

\*OPT 4,n controls the auto-start, governing the action taken when SHIFT-BREAK is pressed, where

n=0 - Cancel start up option

n=1 - \*LOAD a file called \$.!BOOT

n=2 - \*RUN a file called \$.!BOOT

n=3 - \*EXEC a file called \$.!BOOT

\*OPT 7,n is a non-standard command which has been provided to overcome a limitation imposed within the 'Inter' series of applications by Computer Concepts. This facility will only be of relevance if you use a micro fitted with Econet.

For the reasons given below it has been necessary to give CPFS a default filing system identity number of 5. (Every filing system has an identity number. It is the number returned by the MOS call "OSARGS" when it is entered with A=0 and Y=0.) It is important that no two filing systems in the same machine have the same identity number (the MOS would get rather confused), but number 5 is also the network filing system identity. \*OPT7,n allows the filing-system identity of

CPFS to be changed from its current value, where n is the value to be returned by future "OSARGS" calls. (\*OPT7,0 will reset the identity number to the default of 5.)

The applications in the Inter series display the catalogue of the current directory when files are loaded or saved using the menu options. However, for reasons known only to themselves, Computer Concepts decided at the development stage that only filing systems with an identity of 4 (DFS), 5 (NET) and 8 (ADFS) would ever be capable of storing files in directories. The result is that if any filing system responds to OSARGS with any other identity than one of these three values the Inter applications will not attempt to read the catalogue, displaying instead a blank screen.

Since relatively very few machines are used on a network, but (virtually) all 512 users will have and use both DFS and ADFS, the only remaining (ie. mostly unused) filing system identity which can be employed successfully with Inter applications is 5. This then is the default value returned by CPFS.

For micros which are not fitted with Econet this is an unused identity, and Inter applications will 'think' they are loading files from or saving to a network fileserver and will work normally. The vast majority of users can therefore ignore this subterfuge and will have no need to use \*OPT7.

If you are using CPFS in a machine on a network, but are not using any of the Inter series, then you can use \*OPT7 to give CPFS a new identity number. You must select CPFS before logging on to the network and enter:-

\*OPT 7,n

with "n" replaced by your chosen new identity number. Any number between 1 and 15 will be accepted, but values below 11 are best avoided to make sure there is no possible clash with any other filing system.

N.B. For each \*OPT7 command in a Master you must also force the MOS to re-read the current filing system identity by pressing BREAK, because in the Master the MOS stores the current filing system identity and responds to OSARGS itself with the stored value recorded at the last change or reset. On the Master the \*OPT7 value is stored in the

battery-backed CMOS RAM, so once you have made the change you can forget about it, since it will still be the same next time the machine is switched on. In Model B and B+ micros the filing system is checked directly each time the call is issued, therefore this extra step of pressing BREAK is not necessary as \*OPT7 takes effect at once. On these machines, though, you will have to go through this routine every time you switch on.

For those who wish to use Inter applications and the CPFS memory disc on a micro fitted with Econet, an obvious conflict arises. You really have three choices (or some combination of them):

1) You could use \*OPT7 to give CPFS an identity number of 11 or above. All filing systems will then work properly, but the Inter applications will not generate the normal catalogue display when CPFS is the current filing system. This means that you must always enter filenames at the keyboard, there will be no warning if you are going to over-write a file, and loading and saving will be

much slower.

2) If your machine is a Master, you could \*UNPLUG either DFS or ADFS, and then use \*OPT7 to give CPFS an identity of 4 (if it was DFS) or 8 (if it was ADFS). If you \*UNPLUG the ADFS ROM, then you will have to \*INSERT it again before using DOS-Plus. If DFS is \*UNPLUGged, then you will not be able to use the MOS sideways RAM commands \*SRLOAD, \*SRSAVE, etc.

3) You could leave the CPFS identity number as 5 (the same as the network) and trust to luck! Much will still work correctly, but \*TRANSFER will certainly not, and you will not be able to use temporary filing systems or a library filing system on the Master (see below).

## Inactive Commands

-----  
The following commands are recognised by CPFS to provide compatibility with other systems and commercial software packages. They will not cause an error if issued, but because they are logically meaningless they have no effect in CPFS:-

\*BACK

\*CDIR

- \*COPY
- \*DISMOUNT
- \*DRIVE
- \*ENABLE
- \*MAP
- \*MOUNT
- \*TITLE

## USING CPFS WITH OTHER FILING SYSTEMS

=====

Because of its nature (ie. it disappears when you switch off and therefore never contains files when you first start it) it is unlikely you will very frequently use CPFS in isolation. Commonly it will be used in conjunction with another filing system, such as DFS or ADFS, using the disc filing system for permanent storage of files, with CPFS temporarily storing data or programs during the work session. Model B/B+ users will simply switch between CPFS and other filing systems as they are required. Remember though, you do not need to change filing system to use \*TRANSFER, it may be used from any current filing system.

The Master, on the other hand, provides facilities that make it particularly easy to use more than one filing system at once. The method is to have one of them as the current filing system and, when required, to specify another as a temporary filing system as part of the filename or in \*-commands. This is explained briefly in Section G.1 of the Master Reference Manual. The temporary filing system name -CPFS- is used to indicate the Co-Processor Filing System

By including a temporary filing system name in a file-name, a file can be saved to or loaded from any filing system without selecting that filing system as the current one. Thus, whatever filing system is current

SAVE "-CPFS-NEWFILE"

will save your BASIC program as a file called NEWFILE in CPFS.

If you are using data files on the Master, then the same technique can be used to open a file in another filing system. When a file is opened in this way the MOS will make sure the correct filing system is used whenever you use that file's file-handle. For example, suppose in

BASIC you use:-

```
X%=OPENUP("AFILE")
```

to open a file in ADFS called AFILE and you then change to CPFS. If you later use (say)

```
BPUT#X%,50
```

(without interfering with X% yourself, of course) the MOS will make sure that the byte is written to the file AFILE in ADFS. You need not yourself select ADFS; the MOS 'knows' that the file handle X% belongs to ADFS and will activate the correct filing system automatically when it is required. CPFS will remain the current filing system throughout, unless you again change it yourself.

You can also use temporary filing system facilities to issue \*-commands to other filing systems than the current one. One thing to remember here is that there are two kinds of filing system \*-command.

Some filing system commands are recognised and processed by the MOS, so these are common to all filing systems and are called "common" commands. Other commands are recognised and processed by the filing systems themselves, and these are known as "intrinsic" commands. This can be significant if you want to include a filing system name in a \*-command. If the command is a common one, then it does not matter whether the command or the filing system name comes first, but if the command is intrinsic the filing system name must come before the command. Thus from ADFS, for example, you can use

```
*CAT-CPFS- or
```

```
*-CPFS-CAT
```

to get the catalogue of the memory disc, because \*CAT is a common command. However, also from ADFS, you can use

```
*-CPFS-FREE
```

to read the amount of free space in the memory disc, but

```
*FREE-CPFS-
```

will not work correctly because \*FREE is an intrinsic command.

Again on the Master, it is also worth considering the use of a "Library Filing System" if you have a number of machine code utilities on one filing system which you would like to access from another. The idea of the library filing system extends that of a library directory. If you try to \*RUN a machine code program that is not in the current or library directory of the current filing system, the MOS searches for it in the Library Filing System (so long as you have set one).

Remember that the syntax of the LIBFS command is rather odd. You must put the filing system name first as though LIBFS were an intrinsic command, although it is in fact a common one. Thus to set CPFS as your library filing system, you must enter:

\*-CPFS-LIBFS

What all this means in practice is that, in most applications, you can treat the CPFS memory disc just like another disc, but with the name "-CPFS-" rather than ":0", ":1", etc.

Note that only the Master can use temporary or library filing systems. On the Model B you must make sure that the correct filing system is selected before issuing any of its commands or accessing any of its files. Indeed, even on the Master this may sometimes be quicker than using temporary filing systems, especially if ADFS is your current filing system. This is because the current filing system is re-selected on every \*-command. With ADFS this can be a quite lengthy process, depending on the operation to be performed.

## CPFS ERROR MESSAGES

=====

The following errors may be generated by CPFS. They are listed in numerical order with a note explaining the cause or meaning.

No 80186 (Error number 143)

This comes from an attempt to run CPFS on a system without a live 80186 co-processor. The error is generated if no co-processor is fitted (or it was not switched on at the last BREAK), or if one is fitted but it is not an 80186.

Bad Tube configuration (Error number 144)

This occurs if you attempt to select CPFS (eg. \*CP.) with the Tube configured "On".

Outside file (Error number 183)

This results from attempting to set the file pointer beyond the end of a read-only file or of a file which is locked.

Catalogue full (Error number 190)

This results if you attempt to store more than 127 files in CPFS at any one time. In practice this is not likely to be a limitation, since it implies an average file length of only 8K even in a one megabyte 512.

Too many open files (Error number 192)

This happens if you attempt to open more than 5 files simultaneously.

File read only (Error number 193)

This results from an attempt to write to a file that has been opened for input only.

File open (Error number 194)

Various actions (such as deleting) cannot be performed on a file if it is open. This error will result if you try one of them.

Locked (Error number 195)

This happens if you try to update a file that has been locked using \*ACCESS.

Already exists (Error number 196)

This occurs if you try to rename a file using the name of another file that already exists.

CP memory full (Error number 198)

This means that the co-processor memory has been filled and there is no room to save or extend the file.

Bad option (Error number 203)

This will happen if you use a \*OPT command with parameters outside the range used by CPFS.

Bad name (Error number 204)

You will get this error if you specify a file name of zero length or longer than 10 characters, or one that contains any of the characters

# \* : . " - (In some commands (e.g. \*WIPE) # and \* are used as wild cards, in others (e.g. \*RENAME) they will cause this error.)

Bad directory (Error number 206)

Any attempt to use a directory name that is not a valid single character will result in this error.

Bad attribute (Error number 207)

This arises with \*ACCESS if you enter anything after the file specification other than "L".

Not found (Error number 214)

This means that the given file does not exist, or that no file matches the given ambiguous file specification.

Channel (Error number 222)

This will happen if you attempt to use a file handle that has not been allocated to an open file.

EOF (Error number 223)

This happens if you try to read beyond the end of an open file (but see the note under "Technical Information").

Transfer error (Error 224)

This will only occur if one of the filing systems involved in a \*TRANSFER operation has had trouble opening a file.

Bad filing system (Error number 248)

This will occur while using \*TRANSFER if you enter an unrecognised filing system name, or a filing system which cannot be used for some reason (such as a destination filing system name or identity that is the same as the source).

Duplicate filing system ID (Error number 249)

This will occur in \*TRANSFER if you enter a filing system with identity number the same as that set for CPFS. It is only likely to happen if you are using a network. See the section on \*OPT7 if this error occurs. Note that if this error is not generated then that does NOT guarantee that the filing system numbers are valid, since another filing system might share the same ID as CPFS.

TECHNICAL INFORMATION

=====

The system needs one page of private workspace to operate. This means that in a Model B it will raise the value of PAGE by &100.

The Co-processor Filing System uses file handles between &C0 and &C4 (inclusive). Up to 5 files may be open at one time.

The Filing System identity number (as returned by OSARGS with A=0 and Y=0) is 5 by default (ie. it pretends to be NET. See notes for \*OPT7.) CPFS can be started up by means of paged ROM service call &12 with 5 in the Y register. Thus \*FX143,18,5 will make it the current filing system, and the equivalent of this may be a simpler way of starting the system from assembly language than issuing \*CPFS.

All MOS filing system calls in machine code work for CPFS in the same way as they do for DFS and ADFS. These are described in the Reference Manual or User Guide. There are no special OSWORD calls. Some of the calls to OSGBPBPB are specific to the filing system, especially those connected with reading drive numbers, etc. Drive numbers are irrelevant in CPFS, but for compatibility with other systems CPFS responds to these calls as follows:

OSGBPBPB with A=5: For a disc system this reads the disc title, option and drive. For CPFS there is no title (it has length 0) and 0 is returned as the drive number. Thus three bytes are returned. The first and third are always 0 and the second is the option value.

OSGBPBPB with A=6 or A=7: This should read the current (or library) drive name as an ASCII string and the directory name. In CPFS there is no drive, but it is given the dummy name of "0". Directory names are always of length 1, of course. Consequently, four bytes are returned by these calls. The first and third are always 1, the second always &30 (the ASCII value of "0"), and the fourth byte is the name of the current or library directory.

OSGBPBPB with A=9: This is not an Acorn standard and is extra to CPFS. It operates exactly like OSGBPBPB with A=8, except that it reads all file names, not just those in the current directory.

As with DFS, only bit 3 of the least significant byte of the file attributes is relevant. This is set if the file is locked (i.e. "L" attribute is set). Further attributes can be written (using OSFILE 4)

or read (using OSFILE 5), but all bits except bit 3 are ignored. The command \*TRANSFER preserves attributes when transferring files between any of CPFS, ADFS and DFS. For compatibility with ADFS, CPFS files are created with bits 0 and 1 set so that the R and W attributes will be set after the file is transferred to ADFS.

CPFS handles end-of-file conditions in the same way as other BBC Micro filing systems. When using OSBGET (or BGET# from Basic) the first attempt to read beyond the end of the file returns with the C flag set (and EOF# TRUE) but does not generate an error. Subsequent attempts give an "EOF" error (number 223).

-----

Co-Processor Filing System and CPFS are copyright (C) 1991 Essential Software

Essential Software, P.O.Box 5, Groby, Leicester LE6 0ZB