

About MOS Plus

MOS Plus is a ROM specifically designed to plug the gaps in the Master Operating System (Version3.10), fixing the known bugs and making the Master much easier to use. It isn't a Utility ROM; there are many such on the market. It was born out of the frustration of using a Master128 in and office, for normal wordprocessing, spreadsheet and other work, with the usual chore of backing up and renaming files, and all the other things you often do with your machine.

MOS Plus fixes the infamous DFS bug, which writes the wrong length back to an open file when using *CLOSE or CLOSE#0, provides format, backup and verify facilities for ADFS discs and provides a host of minor improvements. Also included is an Alarm Clock system which can be set to beep at a specified time in the future, (or every hour, minute or second) and a help text showing you how to use EDIT,VIEW,VIEWSHEET and TERMINAL.

The idea is that you install MOS Plus and forget it, knowing that many of the problems often faced when using the MASTER will just disappear. MOS Plus can be used in any ROM socket, and so can run in Sideways RAM (from a disc), or as a ROM cartridge, or (the recommended method) in the single spare internal ROM socket of the MASTER.

MOS Plus can be used with Econet, any processor and with virtually any peripheral. Only one restriction exists; the ADFS commands do not work with a SCSI device such as Winchester drive. You should of course find that formatting a Winchester is a fairly rare occurrence.

Installing MOS Plus

This is a fairly simple job for non-technical users and an absolute doddle for anyone who has done it before.

The ROM version

There is one ROM socket inside your Master128 which can be used to hold programs in this form. To fit the ROM, take the lid off the MASTER. On the right of the circuit board are three sockets normally empty, which sit in a row from top to bottom. Gently insert the ROM into the middle one, with the notch pointing to the left. Switch on the machine and look for the '+' sign after the Acorn MOS message. If that's there, everything is alright and you should switch the computer and replace the lid.

MOS Plus may also be fitted to a Master cartridge.

The Disc Version

This is even easier. Just press SHIFT and BREAK and MOS Plus will load into Sideways RAM bank 7. To load into another RAM bank instead, type

```
*SRLOAD MOSPLUS 8000 <bank number>
```

where <bank number> may be 4,5,6 or 7. You must then press CTRL/BREAK to initialize the ROM. The features of MOS Plus are then available in exactly the same ways as the ROM version. Note that the *BACKUP command must not be used with the 'S' option, if you are using the disc version of MOS Plus, as this corrupts the Sideways RAM.

Enhanced Star Commands

Here is a list of the improvements. There are no commands removed and every command can be used as before if you like.

```
*APPEND <fsp>
```

This command now accepts 256 characters from the keyboard instead of 128. To generate the characters above 128 from the keyboard, function keys must be used. For example Teletext control²codes can now be included in text files generated with *BUILD and *APPEND. This feature appears on the MASTER COMPACT.

*BACKUP<source> <destn> (<S>) (<P>) (<Y>).....RON's note also ADFS

Makes a copy of <source> disc onto <destn>. Options:-

S Uses Sideways RAM only to make the backup

P Pauses between reads & writes (Used to copy side 0 of one disc to side 2 of another using a single drive)

Y Bypasses the Y/N check

*BASIC@<Hex addr>

This is extended so that the line tokenizer in BASIC can tokenize text from any address in memory, not just the contents of EDIT's buffer. For example, a BASIC program can be written in VIEW, SAVE'd and then *LOADED at &1000. Issuing *BASIC @1000 would tokenize the file ready to LIST and RUN.

*BUILD<fsp>

Like *APPEND, this now also accepts 256 characters when creating files.

*EDIT

This now correctly accepts the filenames immediately after the command, as in *ED.FILE or with many intervening spaces as in *EDIT FILE. Previously, exactly one space had to be used.

*ROMS

This now list Sideways RAM banks as RAM not ROM and does not list duplicate ROMs as implemented on the Master Compact.

SHOW(<key number>)

This now allows the key number to be omitted and the contents of all programmable keys are shown.

*SRLOAD<fsp><sram address><bank>(<Q>)(<I>)

The command now has an additional optional parameter, 'I', which, after the loading has taken place, copies the byte at &8006 in that bank, into &2A1+bank number. If the file loaded was a ROM image, this has the effect of immediately initializing it so that commands can be issued. This is also found on the Master Compact.

*SRWRITE<addr><addr><sram addr><bank>(<I>)

Like *SRLOAD This will copy a byte after the data transfer. BOTH COMMANDS CAUSE HAVOC IF THE DATA IS NOT A VALID ROM IMAGE.

*STATUS(<item>)

This command now lists the MOS status in alphabetical order, as on the Master Compact.

*UNPLUG<bank>(<I>)

The command now allows ROMs to be UNPLUGed immediately, not on the next CTRL/BREAK , by following the bank number with the letter 'I' (for

immediate).

New Star Commands *BLIST<fsp>(<LISTO format>)

This will take a BASIC program <fsp> and display it on the screen, exactly as with the LIST command. This is useful, say, in the middle of word processing, when you want to check a BASIC file. Without leaving what you are doing, you can check any listing. The command works to full Archimedes BASIC specification, and can decode all BASIC tokens up to BASIC V. The <LISTO format> option will take any number from 0-31, being bit significant codes for listing formatting where, if set:-

b0 A space is inserted between the line number and line text.
b1 Program structures (FOR...NEXT &c.) are indented.
b2 Split the line at a colon
b3 Don't display line numbers
b4 List keywords in lower case

*CATALL(<drive>)

This new command works which only works in ADFS, will catalogue all the files on an ADFS disc, correctly indenting directory levels. Directories are marked by following the name with a letter 'D'.

*DRIVE(<drive>)

This ADFS command is provided to simulate the DFS equivalent for software compatibility. *DRIVE1 is the same as *DIR:1.\$

*EXALL(<drive>)

This new command is like *CATALL, but performs a *EX command on all files. Directories are similarly indented.

*FIND(<:drive><afsp>) (<D>)

This will find a file anywhere on an ADFS disc, and display the filename and full directory path. It's very useful if you save a file on disc but can't remember where. If the command is followed by 'D', the directory names which match the <afsp> are printed. Wild cards '#' and '*' are supported.

*FORMAT<drive><S!M!L>(<Y>) (<N>)

This built in formatter will save a lot of time, as you can now *FORMAT disc without loading the Welcome disc formatter. The parameters are:-

S 40 track single sided disc
M 80 track single sided disc
L 80 track double sided disc

The command responds by asking you to confirm the instruction, by typing 'YES'. This can be bypassed by adding a 'Y' parameter to the commands. Verification after formatting³ is normally automatic, but if you are in a hurry, you can bypass this by adding a 'N' parameter. NOTE if using both options, you should type 'YN' not 'NY'.

*SETTIME<date!time!date and time>

This command sets the date and time, using the same syntax as TIME\$ in BASIC. Of course this can be used from within any language. The syntax is:-

*SETTIME "Wed,11 Nov 1987.11:11:00" or
*SETTIME "WED,11 Nov1987" or
*SETTIME "11:11:00"

*SRKILL<bank id>

This command will wipe (the first 16 bytes of) one or more Sideways ROM images from memory, giving a quick way of clearing out ROM images so that SRDATA can be used. To clear out banks 4 and 7 only, the syntax would be:-

*SRKILL W Z

*VERIFY(<drive>)

This will verify and ADFS disc. If no drive number is given the currently mounted drive is verified.

Alarm Clock System

The Master128 contains an Alarm Clock facility as part of the CMOS RAM chip, which is not used by the Operating System. MOS Plus makes use of the facility to provide a permanent alarm system, which can be set to ring every second, minute or hour, or at a specified time. Two volume settings are allowed.

To use the clock, a link on the main circuit board must be set. The link. LK4, is a jumper with 2 pins. The link is marked on the board and is near the middle of the lefthand edge of the main PCB, to the right of the lefthand coprocessor connector. Therefore, any coprocessor present, must be removed. The pins should be connected with using a jumper link, obtainable from any electronics shop, or in case of difficulty direct from Dabs Press..free on receipt of an S.A.E. Some Master boards don't have the pins present, in which case you will have to join the 2 pads with a small wire link, and solder them together. Again, if in doubt, consult Dabs Press. If you're not sure what you are doing, don't attempt the job. Dabs Press cannot be held responsible for any damage to machines.

Turn your machine off to set the link, and turn it on again. To test the alarm, make sure MOS Plus is installed, and type *ALARM S. The machine should start beeping every second. The commands available under the alarm system are:-

*ALARM Prints alarm setting on the screen. If the alarm is set to an illegal time, asterisks will be printed.

*ALARM<time> Sets the alarm. The time must be given in the format <HH:MM:SS>. The alarm will then ring at the specified time, for about 90 seconds. To terminate the sounds, type *ALARM OFF (permanently stopped) or press BREAK (stopped until the same time tomorrow)

*ALARM S Sounds the alarm every second. Use ALARM M for every minute on the minute and *ALARM H for every hour on the hour

*ALARM ON Enables the alarm

*ALARM OFF Disables the alarm

*ALARM QUIET Alarm will ring with low volume. This can be permanently set in CMOS RAM with *CONFIGURE ALARM QUIET.

*ALARM LOUD Alarm will ring with high volume. This can also be permanently set with *CONFIGURE ALARM LOUD

Once the alarm is set it will survive 'reset' and 'power down'

When the alarm is sounded, a user event (No.9) is generated. This can be

picked up by setting the event vector to point to your routine, and at the end of your routine, jump to the address represented by the old contents of the vector. Your routine (which should only work if the accumulator equals 9 on entry), will be called every time the alarm signal occurs. Additional event entry values are provided in the 'X' register as follows:-

```
0 Alarm at specified time
1 Second time signal
2 Minute time signal
3 Hour time signal
```

If on exit from the event routine, you set the top bit of X, then the alarm bell will not sound.

Mouse Driver

Many software packages offer mouse facilities, which usually require to have an AMX or similar ROM fitted. The code to convert mouse movements into co-ordinate data is contained in MOS Plus, allowing access to many mouse driven packages. OSWORD &64 is used, and is called without entry parameters. On return, the p'rameter block contains data as follows:-

```
XY,XY+1      X Co-ordinate (0-1279)
XY+2,XY+3    Y Co-ordinate (0-1023)
XY+4         X text co-ordinate
XY+5         Y text co-ordinate (only valid for 32 line modes)
XY+5         Buttons bit 4-0 always zero
              Bit 5 Right-hand button 0=preserved 1=Not preserved
              Bit 6 Middle button 0=preserved 1=Not preserved
              Bit 7 Left-hand button 0=preserved 1= Not preserved
```

Mouse Driver

Many software packages offer mouse facilities, which usually require you to have an AMX or similar ROM fitted. The code to convert mouse movements into co-ordinate data is contained in MOS Plus, allowing access to many mouse driven packages. OSWORD &64 is used, and is called without entry parameters. On return, the parameter block contains data as follows:-

```
XY,XY+1      X Co-ordinate (0-1279)
XY+2,XY+3    Y Co-ordinate (0-1023)
XY+4         X text co-ordinate
XY+5         Y text co-ordinate (only valid for 32 line modes)
XY+5         Buttons bit 4-0 always zero
              Bit 5 Right-hand button 0=preserved 1=not preserved
              Bit 6 Middle button 0=preserved 1=Not preserved
              Bit 7 Left-hand button 0=preserved 1= notpreserved
```

Bug Fixes

MOS Plus fixes two major bugs in the Master Operating System. The first is the infamous *CLOSE bug where DFS files do not close with the correct length if *CLOSE or CLOSE#0 is used. This now works correctly. The second is that *EDIT now accepts any number of spaces between command and filename. Some improvements have been made. *REMOVE now fails if a second filename follows the first. This stops problems with using *REM. when you meant *REN. for RENAME. The CMOS RAM R-Poweron reset now puts sensible numbers in the CMOS bytes, unlike the previous situation where all parameters were set to zero.

Help System Version 1.09

1. *CLOSE now keeps an error pointer
2. Service &25 is no longer claimed, allowing MOS Plus to work with the *CONVERT ROM on the Welcome disc.
3. *VERIFY no longer crashes if an error is generated when running on a second processor.

Version 1.12

1. *FIND now gives account of the objects found at the end of the search, and now takes an optional search parameter.
2. *FIND takes an optional 'F' parameter [*FIND<list spec> (D)(F)] which lists full catalogue information, as in *EX, rather than just the filename. If the 'D' parameter is used, the 'F' must not precede it.
3. *FORMAT no longer corrupts main memory, and the bug which caused an intermittent crash on second processors when trying to format a write-protected disc finally removed.
4. *BACKUP with Sideways RAM now 5-10 times faster.
5. ADFS errors now proper SCSI codes not 1770 codes when Winchester in use.
6. DFS *BACKUP reports 'Write-protected disc' and not 'Disc error 12'
7. MOS+ no longer claims a page of workspace in HAZEL.

Version 1.13

1. *FIND no longer fails if the required file is in a directory whose name is exactly 10 letters long.
2. The ROM now works with MOS 3.23 and 3.26 (unreleased as at 25th Feb '88)
3. *FIND now correctly selects screen memory at all times.
4. *SRKILL now clears 127(not 16) bytes of a bank, to avoid problems with long title strings.

Version 1.14

1. Quite a few new items are added in version 1.14. The mouse driver is now removable to avoid conflict with other mouse drivers such as Nidd Valley Chaffeur. A start string can be set which will be executed rather like a !BOOT file on power-up and CTRL/BREAK. The screen colour can be set (if not starting in mode 7) and the 50 bytes CMOS data can be loaded and saved to disc, useful if you need to change the batteries. *ECHO works in the same way as the Archimedes command of the same name-you can perform VDU sequences wherever you can issue * commands e.g. *ECHO !S!@!D!@!@!@ will create a blue screen. A printer specified command *LPRINT will send data to the printer only, so *LPRINT ![E will set bold type (the ![command is equivalent to ESC)

1. *CONFIGUREMOUSE and *CONFIGURENOMOUSE will enable or disable the mouse driver at the next hard reset.
2. *CONFIGURESTART(<string>) Sets string to be pushed into the keyboard buffer on hard resets. The string should be GS format (i.e !G=CHR\$(7) etc.) and truncated to 20 characters. For example:-
*CONFIGURESTART*CAT!M will make your Master catalogue a disc every time it is switched on, or CTRL/BRK is pressed. To cancel any start string type *CONFIGURESTART and *STATUS will show 'No Start'. With this command and a mains timer, you can start automatic tasks.
3. *CONFIGURECOLOUR<0-15> Sets background colour to be forced (by VDU19) on each hard re-set. Does not work if your start-up mode is 7 or 135.
4. *LCONFIG <filename> will load 50 bytes of CMOS RAM. *SCONFIG<filename> will save 50 bytes of CMOS RAM.
5. *ECHO<string> sends a GS format string to the current output stream.
*LPRINT <string> sends the GS format string to the printer only (uses VDU1)

The new *CONFIGURE commands all appear in the correct alphabetical place when listed with *STATUS. This is because MOS Plus entirely replaces the *STATUS command.