PC-MOS 5.01 Booster Version 1.00

PC-MOS 5.01 Booster Version 1.00 supplies significant enhancements over previously available versions of PC-MOS. The PC-MOS 5.01 Booster also includes all maintenance updates to the original version of PC-MOS 5.01. These enhancements and updates are detailed below.

The PC-MOS 5.01 Booster can be used with any update level of PC-MOS Version 5.01.

Table of Contents

Quick Start

1. If your PC-MOS configuration does not include memory management, or you want to install the PC-MOS 5.01 Booster to a PC-MOS floppy boot disk, see the relevant installation instructions below.

2. Only task 0 may be active when installing PC-MOS 5.01 Booster. If necessary, after exiting any active applications in each task, enter REMTASK ALL from the task 0 PC-MOS command prompt.

3. Since critical PC-MOS boot files will be changed, the INSTALL program requires that disk caching be turned off in case the PC-MOS 5.01 Booster installation process is interrupted before completion. Enter MOSADM CACHE OFF at the PC-MOS command prompt to turn disk caching off. Note that disk caching will need to be turned off even if you have no CACHE= line in your CONFIG.SYS.

4. Since critical PC-MOS boot files will be changed, we recommended that you have a PC-MOS floppy boot disk available in case the PC-MOS 5.01 Booster installation process is interrupted before completion. This is also a very good time to make sure that you have a current backup of the drive onto which you are installing PC-MOS 5.01 booster.

5. The PC-MOS 5.01 Booster is installed from any drive by running the INSTALL.EXE program supplied on the PC-MOS 5.01 Booster diskette. For example, if you inserted the PC-MOS 5.01 Booster diskette into drive B:, you would enter B:INSTALL at the PC-MOS command prompt.

6. If, for any reason, you need to recover your original PC-MOS configuration, you can uninstall PC-MOS 5.01 Booster using the same INSTALL program by selecting Uninstall from the INSTALL main menu.

Installing with no Memory Management

The INSTALL program requires about 450K free memory. If you temporarily change your CONFIG.SYS file so your SMPSIZE is 14K, then reboot, you should have enough free memory to install PC-MOS 5.01 Booster. This is a large enough SMP to load UPDAT501.SYS, but is probably not large enough for any other device drivers. After installing PC-MOS 5.01 Booster, you can restore your original CONFIG.SYS file.

If your original CONFIG.SYS file did not load UPDAT501.SYS, be sure to add a DEVICE= line to your restored CONFIG.SYS so that UPDAT501.SYS is loaded. The installation will have added the correct DEVICE= line as the last line in your temporary CONFIG.SYS file. For example, if you installed PC-MOS 5.01 Booster into C:\PCMOS, you would include a DEVICE=C:\PCMOS\UPDAT501.SYS line in your restored CONFIG.SYS, exactly like the last line in your temporary CONFIG.SYS file.

Installing to a PC-MOS Floppy Boot Disk

After booting from the PC-MOS floppy boot disk, insert your PC-MOS 5.01 Booster diskette into any different drive. You can then install PC-MOS 5.01 Booster onto your PC-MOS floppy boot disk by running the INSTALL.EXE program supplied on the PC-MOS 5.01 Booster diskette. For example, if you booted from a PC-MOS floppy boot disk in drive A: and inserted your PC-MOS 5.01 Booster diskette into drive B:, you would enter B:INSTALL at the PC-MOS command prompt.

$$SHELL.SYS/COMMAND.COM/ADDTASK.COM
Create larger tasks to run larger programs

Two styles of $$SHELL.SYS and COMMAND.COM are provided with PC-MOS 5.01 Booster. The NEW style is installed when you install to a hard disk in a memory-managed PC-MOS configuration, otherwise the OLD style is installed. Use the NEWCMD.BAT batch file to switch to the NEW style. Use the OLDCMD.BAT batch file to switch to the OLD style.

The OLD style, implemented by $$SHELL.OLD and COMMAND.OLD, is the same as the style of the $$SHELL.SYS and COMMAND.COM you've been using with PC-MOS. $$SHELL.SYS implements the bulk of the command processor. It is loaded into global memory where it is accessible to all tasks. COMMAND.COM is a small stub serving primarily to reserve task memory for the task-specific data areas used for processing commands. $$SHELL.OLD and COMMAND.OLD are installed in your PC-MOS system directory. They were copied onto the active copies of $$SHELL.SYS and COMMAND.COM by the INSTALL program if needed.

ADDTASK.COM has been changed so that a task which is only large enough to hold COMMAND.COM can be created. ADDTASK 0 will report the minimum task size according to your configuration. In a memory-managed configuration, with the OLD style, this minimum task size is 7K. Note that the minimum task size will not allow you to run any programs, you will only be able to use internal command processor commands like DIR.

The NEW style is implemented by $$SHELL.NEW and COMMAND.NEW which were installed in your PC-MOS system directory. These are copied onto the active copies of $$SHELL.SYS and COMMAND.COM by the INSTALL program. $$SHELL.NEW is a tiny stub program consuming only 16 bytes of global memory as $$SHELL.SYS. All the remaining PC-MOS command processor functionality is in COMMAND.NEW, which is loaded into each task as COMMAND.COM.

The NEW command processor loads most of itself into upper task memory, leaving a small, resident stub in low task memory. When you run a program, the command processor unloads all but the resident stub, freeing most of its memory for the use of the program you are running. When the program terminates, the resident stub reloads the remainder of the command processor, but only if the program actually used the freed upper task memory.

(continued)

The advantage of the NEW style occurs primarily in two configurations -- in a memory-managed configuration and in a non-memory-managed, single-task configuration.

With the NEW style, in a single-task configuration, the maximum task size is 33K larger than with the OLD style. In itself, this doesn't represent a memory savings, we've just moved the command processor from global memory to task memory. The savings occurs when you run a program. In the NEW configuration, much of the command processor's memory becomes available to the program you've run. In the OLD configuration, the memory was always used by the command processor. You will find that your program has 30K more memory available to it in this non-memory-managed, single-task configuration.

In a memory-managed configuration, the stub $$SHELL.SYS will allow 33K of global memory to be used for some other purpose, with very little cost in the amount of memory available to application programs. This could permit a larger SMP or larger CACHE but, more likely, it allows more of your existing configuration to fit in upper memory. If 33K more of your existing configuration fits in upper memory, 33K more memory is available for larger task sizes. In that case, as in the single-task configuration described in the preceding paragraph, 30K more memory is available to your application programs.

The NEW style has disadvantages in two configurations -- in a non-memory-managed, multi-tasking configuration and in any configuration where the reloading of the command processor from disk is a problem. Reloading the command processor from disk is probably only a disadvantage in non-memory-managed, floppy boot configurations. In a memory-managed configuration, you could use $RAMDISK.SYS to create a ram disk to hold COMMAND.COM and use SET COMSPEC=d:\COMMAND.COM (where d: is the ram disk drive) so the command processor is reloaded from the ram disk.

In a non-memory-managed, multi-tasking configuration, the OLD style should be used.

==========

$MEMSEL.SYS

One boot disk serves memory-managed 286 and 386 systems

$MEMSEL.SYS allows boot-time selection of a MEMDEV driver according to whether the boot CPU is a 286 or is a 386 or better. This allows a single boot disk (or boot configuration) to be used for both environments. The $MEMSEL.SYS command line is:

MEMDEV = $MEMSEL.SYS [/286=d:\path\286driver] [/386=d:\path\386driver]

/286=$GIZMO.SYS /386=$386.SYS are the defaults when /286= or /386= are not specified.

==========

SMPDUMP.EXE
Tune your SMPSIZE to free more memory

SMPDUMP.EXE dumps all memory blocks in the SMP(s). This information allows you to tune your SMP sizes to your exact, actual usage. For the first time you can determine the precise cost in your configuration of adding a task or of running an application. If this tuning allows you to reduce your SMP sizes so that all your kernel components are loaded in upper memory (as reported by MOS INFO), your maximum task size could be increased to 612K with $386.SYS (708K if your hardware and requirements permit the CONFIG.SYS command VTYPE=4F to be used). A sample dump is:

```
C:\>SMPDUMP
SMPDUMP System Memory Pool Dump Utility Version 5.01 (940326)
(C) Copyright 1994 The Software Link, Incorporated All
rights reserved worldwide

Segment  Length  Type
   CC00     400  used
   CC19     528  used
   CC3A      96  Block Device Block for drive A:
   CC40      96  Current Directory Block for drive A: for task 0
   CC46      96  Block Device Block for drive B:
   CC4C      96  Current Directory Block for drive B: for task 0
   CC52     128  used
   CC5A      96  Block Device Block for drive C:
   CC60      96  Current Directory Block for drive C: for task 0
   CC66      96  Global File Block for file AUX
   CC6C      48  Task File Block for file AUX for task 0 for PSP 0000
   CC6F      96  Global File Block for file CON
   CC75      48  Task File Block for file CON for task 0 for PSP 0000
   CC78      96  Global File Block for file PRN
   CC7E      48  Task File Block for file PRN for task 0 for PSP 0000
   CC81      96  Current Directory Block for drive A: for task 0
   CC87      96  Current Directory Block for drive B: for task 0
   CC8D    2880  Character device driver _501_134
   CD41    1280  used
   CD91      96  Current Directory Block for drive C: for task 0
   CD97     208  used
   CDA4    3728  Task Control Block for task 1
   CE8D      96  Current Directory Block for drive A: for task 1
   CE93      96  Current Directory Block for drive B: for task 1
   CE99      96  Current Directory Block for drive C: for task 1
   CE9F     208  used
   CEAC      48  Task File Block for file CONFIG.SYS for task 1 for PSP 08D5
   CEAF      48  Task File Block for file AUX for task 1 for PSP 0000
   CEB2      48  Task File Block for file CON for task 1 for PSP 0000
   CEB5      48  Task File Block for file PRN for task 1 for PSP 0000
   CEB8      96  Global File Block for file CONFIG.SYS
   CEBE      32  Record Lock Block for file CONFIG.SYS for task 1 for PSP 08D5
   CEC0      96  Global File Block for file SMPDUMP.OUT
   CEC6      48  Task File Block for file SMPDUMP.OUT for task 0 for PSP 0706
   CEC9   90992  free
```

Enhancements and updates since Update #6

1. $$SHELL.SYS allows rename file and rename directory commands of the form:

   RENAME d:\path\oldname newname

   even when d: is not the current drive. The drive is not specified for the new name, it is assumed to be the same as the drive of the old name.

   $$SHELL.SYS closes any open files it may have inherited (other than the standard files) when it starts up, more closely matching the behavior expected by some applications.

2. $PIPE.SYS now accepts a /B flag which allows binary data to be sent without an appended end of file marker character (1AH).

3. $386.SYS now accepts a /W flag to limit memory testing to the first 16 megabytes. On some machines with Cyrix 386-replacement 486's the CPU addressing is limited to 16 megabytes even though the CPU looks like a 486.

   $386.SYS is more demanding in its tests to see if it can relocate an extended BIOS data area, fixing a problem on Gateway 2000 with a SCSI caching disk controller.

4. MOS.COM now allows MOS USEIRQ and FREEIRQ to specify IRQ's 8 through 15. This requires that UPDAT501.SYS patch 130 be active.

   MOS MAP now displays $'s if present in program filenames.

   MOS DOSVER allows up to 6.xx. MOS DOSVER has been corrected so that version x.x9 is not changed to version x.x0.

   MOS SERINIT now allows specification of port numbers up to 32.

5. UNTERM.SYS now supports the more recent AMR cards. UNTERM.SYS requires that the enabled ROM video BIOS be compatible with all co-resident AMR cards. Thus a co-resident VGA card is only supported if the ROM BIOS on the VGA card is compatible with all co-resident AMR cards' video chip set. Mixed AMR cards are only supported if the enabled video BIOS is compatible with the other AMR cards' video chip set.

6. Patch 5 ensures that the CPU stack pointer is on a word boundary at all times in the MOS kernel. You may measure an improvement of as much as 10% in performance for some activities. This patch is automatically applied during installation of PC-MOS 5.01 Booster.

==========

Update #6 enhancements and updates

1. An updated Auto Configuration Utility ACU.EXE that has been modified to make sure that the UPDAT501.SYS device driver statement is included in your CONFIG.SYS file.

2. A new FORMAT.COM that will not run when booted under DOS 5.0. The master boot record could be overwritten if you used FORMAT.COM to format a MOS secondary partition when running under DOS 5.0. FORMAT.COM will now only run when booted under PC-MOS.

3. A program called SFT.COM that must be run BEFORE you try to install FOXPRO on your system. SFT.COM is a TSR program that allows FOXPRO's install program to "see" the setting it requires for FILES= so that it will function properly.

4. Updated EmuLink terminal device drivers ELTERM.SYS and HGTERM.SYS that contain a correction for printing at the terminal using the MOS ROUTE LPTn TO TERM command. When printing from an application that caused the screen to scroll during the print, extra characters would be sent to the printer. Also, these drivers make sure that caps lock, num lock, and scroll lock work correctly at the terminal. In order for this second correction to work properly, the patch E31_P2.PAT must be manually applied to EmuLink 3.1.

5. The patch E31_P2.PAT that must be manually applied to EmuLink 3.1 to ensure that caps lock, num lock, and scroll lock work correctly at the EmuLink terminal. Instructions on how to apply the patch are in the patch file itself. Use the TYPE command to display the contents of the patch file on the screen.

6. An updated PC-type terminal device driver PCTERM.SYS that contains a correction for printing at the terminal using the MOS ROUTE LPTn TO TERM command. When printing from an application that caused the screen to scroll during the print, extra characters would be sent to the printer.

7. An updated MOSADM.COM file that corrects a problem that caused an error message to display when trying to write to the same COM device from more than one task at a time.

8. An updated MOS System Monitor program (MONITOR.COM). The new program contains a correction for the displayed number of files open. Also, a new /U option is available when loading the program for the first time during a session. Entering MONITOR /U will load the program and immediately display the main menu, as opposed to having to press CTRL - SPACE to bring up the menu.

9. An updated MOS.COM file that contains a revised MOS ROUTE LPTn command option. Entering just MOS ROUTE at the system prompt, with no operands, will display a list of any printer redirection presently set and active on the system.

10. A new $386.SYS memory management driver that was revised to compensate for a known problem in some of the early IBM 386SLC microprocessors used in some IBM PS/2 computers.

(continued)

11. A more efficient $SERIAL.SYS driver that improves communications speed through standard serial ports. Also, input and output FIFO support for 16550A UARTs has been added. This provides an additional increase in serial transmission rates. Two new operands (IF= and OF=) have been added to the end of the $SERIAL.SYS command statement to provide this support, as follows:

```
DEVICE={d:\path\}$SERIAL.SYS  /AD=nnnn,IB=nnnnn,
OB=nnnnn,HS={N,D,X,P,R},IN=n,IF=nn,OF=nn/...
```

where:

IF=nn is the Input FIFO buffer size in bytes. Defaults to 0 - disabled. (Can ONLY be used on serial ports using a 16550A UART. For such ports, set to 14.)

OF=nn is the Output FIFO buffer size in bytes. Defaults to 0 - disabled. (Can ONLY be used on serial ports using a 16550A UART. For such ports, set to 16.)

12. A new SERINFO utility program that can be used to see what serial communication ports are in use on your system and what optional parameters are set for each port. To run the utility, enter the following command at the system prompt:

```
SERINFO
```

13. An updated EXPORT.EXE backup program that contains a new command line operand and corrections for appending an export to an existing EXPORT diskette set and for importing large export files that span multiple diskettes.

New /f Operand:

A new /f operand is available that selects an alternate behavior for the /s copy all subdirectories operand, as follows:

```
EXPORT C:\ A: /S
```

The above command will export all files in the root and all files in all subdirectories.

```
EXPORT C:\STUFF A: /S
```

The above command will export all files within C:\STUFF (presuming that STUFF is a directory rather than a file) and all files in all child directories.

```
EXPORT C:\STUFF\M*.* A: /S
```

The above command will export all files within the C:\STUFF directory that start with the letter "m". It will also find every subdirectory within C:\STUFF whose name starts with an "m" and then export all files within those subdirectories and their child directories.

(continued)

EXPORT C:\STUFF\M*.* A: /S /F

The above command will export all files within the C:\STUFF directory that start with the letter "m". It will also find every subdirectory within C:\STUFF and then export only those files within those subdirectories and their child directories that start with the letter "m".

NOTE: The /f switch cannot be used without the /s switch.

/a APPEND Operand:

The EXPORT append operation has been corrected to properly identify the last diskette of an existing backup set. If an attempt is made to start an append operation with a diskette other than the last diskette of an existing backup set, an error message will be displayed. You will have to start the EXPORT append operation over again with the correct diskette in the drive.

The first diskette used in an append operation (the last diskette of the current backup set) will not be cleared of existing data, but appended to. Any subsequent diskettes used will be cleared of any existing data and overwritten with the new EXPORT data.

/d OPERAND, Specified Date:

The User's Guide says that the /d operand selects files for export with dates that come after the specified date. This is not correct. Files with dates that match or come after the specified date will be selected for export.

EXPORT Files that Span Multiple Diskettes:

A problem has been corrected in which large exported files that spanned more than one diskette could sometimes not be properly restored with the IMPORT program. This was found to be a problem in the EXPORT program and is corrected in the EXPORT.EXE provided with this update.

14. A new UNTERM.SYS driver for Video Network Adapter Plus (VNA Plus) and Video Graphic Network Adapter Plus (VGNA Plus) hardware that fixes problems with printing through the parallel port on workstation Interface Units and random workstation lock-ups.

15. A revised $ARNET.SYS driver for the "SmartPort-8" and "SmartPort Plus-16" intelligent multiple serial port boards by ARNET Corporation.

(continued)

16. Patches that must be manually applied to LANLink 5X and LANLink LapTop to provide compatibility with PC-MOS 5.01. The patch files are:

LL5M.PAT - This patch is REQUIRED for PC-MOS 5.01 and LANLink 5X. The patch is applied to LANSERVE.COM.

LL5MDEMO.PAT - This patch is REQUIRED for PC-MOS 5.01 and LANLink 5X. The patch is applied to LANSERVE.COM on the LANLink 5X DEMO Disk.

LL5XSAT.PAT - This patch is REQUIRED for PC-MOS 5.01 and LANLink 5X. The patch is applied to LANSAT.SYS.

LLLAP5M.PAT - This patch is REQUIRED for PC-MOS 5.01 and LANLink LapTop. The patch is applied to LANSERVE.COM.

LLLAPSAT.PAT - This patch is REQUIRED for PC-MOS 5.01 and LANLink LapTop. The patch is applied to LANSAT.SYS.

The patch files themselves are ASCII text files that contain both the patch and the instructions on how to apply the patch. Use the TYPE command to display the contents of the patch file on the screen or the COPY command to send it to the printer. Then follow the instructions to apply the patch.

(continued)

17. PATCHES FOR PC-MOS:

These patches ARE AUTOMATICALLY APPLIED to PC-MOS by the INSTALL program!

Patch 2: (Patch 1 is replaced by Patch 2)

This patch must be applied to the PC-MOS kernel to correct a possible incorrect partition boot record entry. This problem primarily occurs if you have a secondary partition formatted under PC-MOS 4.10 and then reformat it under PC-MOS 5.01. This will happen when using AutoInstall to upgrade an existing dual boot system. The incorrect partition boot record entry will prevent the system from booting from the PC-MOS secondary boot partition. In some cases this may also happen when formatting a new secondary partition made with PC-MOS 5.01 HDSETUP. In that case, you will get a "Boot Error" message when attempting to boot from the PC-MOS secondary boot partition.

Patch 3:

This patch must be applied to the PC-MOS kernel to correct the method that PC-MOS uses for terminating the System File Table. This is required for PC-MOS security to function correctly in a local area network environment.

Patch 4:

This patch must be applied to the PC-MOS kernel to allow installation of the Bernoulli 90 MB Pro Series removable cartridge system. Without this patch, the DOSOAD.SYS device driver and DOSCFG.EXE configuration program that come with this system will not load and run.

==========