Apple II
Technical Notes

Developer Technical Support

ProDOS 8
#23: ProDOS 8 Changes and Minutia

Revised by: Matt Deatherage September 1990
Written by: Matt Deatherage July 1989

This Technical Note documents the change history of ProDOS 8 through V1.9, and it supersedes the information on this topic in the *ProDOS 8 Technical Reference Manual* and the *ProDOS 8 Update*.
Changes since March 1990: Updated to include ProDOS 8 version 1.9 and its known bugs.

Changes? You're kidding.

No. One of the side effects of evolving technology is that eventually little things (like the disk operating system) have to change to support the new technologies. Every time Apple changes ProDOS 8, the manuals can't be reprinted. For one thing, it takes a long time to turn out a manual, by which time there's often a new version done which the new manual doesn't cover. For another thing, programmers and developers don't tend to purchase revised manuals (our informal research shows that more people have up-to-date Apple /// RPS documentation than have up-to-date Apple IIc documentation—and this was done before the Apple IIc Plus was released…).

So this Note explains what has changed between ProDOS 8 V1.0 and the current release, V1.9, which began shipping with System Software 3.2. Table 1 shows what versions of ProDOS 8 existing documentation covers.

Version

Document	Number
ProDOS 8 Technical Reference Manual	1.1.1
ProDOS 8 Update	1.4
AppleShare Programmer's Guide to the Apple IIGS	1.5

Table 1–ProDOS 8 Documentation

ProDOS 1.0

This was the first release of ProDOS, which was so unique it didn't even have to be called ProDOS 8 to distinguish it from ProDOS 16. If you have documentation that predates ProDOS 1.0, you should seek professional help from APDA at the address listed in Technical Note #0.

ProDOS 1.0.1

- Fixed a bug in the `STATUS` call which affected testing for the write-protected condition.

ProDOS 1.0.2

- Changed instructions used in interrupt entry routines on the global page so the accumulator would not be destroyed.
- Fixed a bug in the Disk II core routines so the motor would shut off after recalibration on an error.

ProDOS 1.1

- Changed the internal MLI layout for future expansibility and maintenance.
- Modified machine ID routines to identify IIc and enhanced IIe ROMs.
- Removed code that allowed ProDOS to boot on 48K machines.
- Removed the check for the ProDOS version number from the `OPEN` routine.
- Incremented `KVERSION` (the ProDOS Kernel version) on the global page.
- Modified the loader routines to reflect the presence of any 80-column card following the established protocol (see ProDOS 8 Technical Note #15, How ProDOS 8 Treats Slot 3). Also, at this time, added code to allow slot 3 to be enabled on a IIe if an 80-column card following the protocol was found.
- Added code to turn off all disk motor phases prior to seeking a track in the Disk II driver.
- Fixed a bug to prevent accesses to /RAM after it had been removed from the device list.
- Reduced the size of the /RAM device by one block to protect interrupt vectors in the auxiliary language card. The correct vectors are installed at boot time.

ProDOS 1.1.1

- Fixed a Disk II driver bug for mapping into drive 1.
- Modified machine ID routines to give precedence to identifiable 80-column cards in slot 3.

ProDOS 8 1.2

- Changed the name from ProDOS to ProDOS 8 to avoid confusion with ProDOS 16, which, again, this Note does not discuss.
- Introduced the clock driver for the Apple IIGS. The machine identification code was changed to indicate the presence of the clock on the IIGS.
- Added preliminary network support by adding the network call and preliminary network driver space.
- Fixed a bug in returning errors from calls to the RAM disk. Changed the RAM disk driver to return values of zero on reads and ignore writes to blocks zero, one, four, five, six, and seven, which are not accessible as storage in the driver’s design.

- Added a new system error ($C) for errors when deallocating blocks from a tree file.
- Fixed a bug in zeroing a Volume Control Block (VCB) when trying to reallocate a previously used VCB.
- Modified the ProDOS 8 loader code to automatically install up to four drives in slot 5 if a SmartPort device is found. Removed the code to always leave interrupts disabled, which leaves the state of the interrupt flag at boot time unchanged while ProDOS 8 loads.
- Changed the MLI entry to disable interrupts until after the `MLIACTV` flag is set and other ProDOS parameters are initialized.
- Modified the `QUIT` code to allow the Delete key to function the same as the left arrow key. Also fixed a bug so screen holes would not be trashed in 80-column

mode. Crunched code to allow soft switch accesses to force 40-column text mode. Fixed a bug so the dispatcher would not trash the screen when executed with a `NIL` prefix.

- Modified the `ONLINE` call so that it could be made to a device that had just been removed from the device list by the standard protocol. Previous to this change, a VCB for the removed device was left, reducing the number of on-line volumes by one for each such device. From this point on, removing a device should be followed by an `ONLINE` call to the device just removed. The call returns error $28 (`No Device Connected`), but deallocates the VCB.
- Added a spurious interrupt handler to allow up to 255 unclaimed interrupts before system death.
- Removed the code which invoked low-resolution graphics on system death—it had not worked well and the space was needed. The system had previously had the ability to display "`INSERT SYSTEM DISK AND RESTART`" without also displaying "`-ERR xx`", which was removed at this point for space reasons since the system wasn't using it (and hopefully you weren't, either, since it wasn't documented).
- Changed `MLIACTV` to use an `ASL` instead of an `LSR` to turn "off" the flag.
- Changed the `OPEN` call to correctly return error $4B (`Unsupported Storage Type`) instead of error $4A (incompatible file format for this version) when attempting to open a file with an unrecognized storage type.
- Fixed an obscure bug involving `READ` in Newline mode. If the requested number of bytes was greater than $FF, **and** the number of bytes in the file **after** the newline character was read was a multiple of $100, then the number of bytes reported transferred by ProDOS was equal to the correct number of transferred bytes plus $100.
- Starting with V1.2 on an Apple IIGS, stopped switching slot 3 ROM space and left the determination of whether the slot or the port was enabled to the Control Panel; however, there was a bug in this implementation which was fixed in V1.7 and described in ProDOS 8 Technical Note #15, How ProDOS 8 Treats Slot 3.
- Updated the slot-based clock driver's year table through 1991.
- Added a feature which allows ProDOS 8 to search for a file named ATINIT in the boot volume's root directory, to load and execute it, then to proceed normally with the boot process by loading the first .SYSTEM file. No error occurs if the ATINIT file is not found, but any other error condition (including the file existing and not having file type $E2) causes a fatal error.
- Changed loader code so ProDOS 8 could be loaded by ProDOS 16 without automatically executing the ATINIT and the first .SYSTEM file.
- Changed the device search process in the ProDOS 8 loader so SmartPort devices are only installed if they actually exist, and Disk IIs are placed with lowest priority in the device list so they are scanned last.
- Forced Super Hi-Res off on an Apple IIGS when a fatal error occurs. (Actually,

this did not work, but it was fixed in V1.7.)

- Inserted a patch to fix a bug in the first IIGS ROM that caused internal $Cn00 ROM space to be left mapped in if SmartPort failed to boot.

ProDOS 8 1.3

Warning: This is not a stable version of ProDOS due to an illegal 65C02 instruction which was added. This version can damage disks if used with a 6502 processor.

- Changed the code that resets phase lines for Disk IIs so phase clearing is done with a load instead of a store, since stores to even numbered locations cause bus contention, which is major uncool. Changed the routine to force access to all eight even locations, which not only clears the phases, but also forces read mode, first drive, and motor off. DOS used to do this; ProDOS had not been doing it. If `L7` had been left on when the Disk II driver was called and it checked write-protect with `L6` high, write mode was enabled. Forcing read mode leaves less to chance.

- Changed deallocation of index blocks so index blocks are not zeroed, allowing the use of file recovery utilities. Instead, index blocks are “flipped” (the first 256 bytes are exchanged with the last 256 bytes).
- Since the UniDisk 3.5 interface card for the][+ and IIe does not set up its device chain unless a ProDOS call is made to it, ProDOS `STATUS` calls are now made to the device before SmartPort `STATUS` calls.

ProDOS 8 1.4

- Removed an illegal 65C02 instruction which was added in V1.3.
- Modified the Disk II driver so a routine that should only clear the phase lines only clears the phase lines. Also clear `Q7` to prevent inadvertent writes.

Warning: The AppleTalk command, which was added in version 1.5, is present as a skeleton in this version. Unfortunately, it’s not a useful skeleton. It moves a section of memory from a ProDOS location to another location and transfers control, totally oblivious of the fact that there is no code at this address.

Even more unfortunate, the server software that ships with the Apple IIe Workstation Card is such that when the IIe is booted over the network with that server software, it is version 1.4 (`KVERSION = 4`).

So if you boot version 1.4 from a local disk, making a $42 call is fatal. See ProDOS 8 Technical Note #21, Identifying ProDOS Devices, for a reliable way to identify AppleTalk volumes under ProDOS 8 version 1.4.

ProDOS 8 1.5

- ProDOS 8 1.5 is the first version to include network support through the ProDOS Filing Interface (PFI) as part of ProDOS 16 or on the Apple IIe Workstation Card without booting over the server (see the warning under version 1.4). Made many changes to internal routines for PFI location and compatibility at this point. Crunched and moved code for PFI booting and accessibility.
- Changed some strings to all uppercase internally for string comparisons.
- Removed the generic $42 AppleTalk call (the cause of the previous warning) which was introduced in V1.2, as PFI gets called through the global page.
- Changed the `ASL` to clear the `MLIACTV` flag back to an `LSR`. This doesn’t make

nested levels of busy states possible, but always clears the flag before calling interrupt handling routines that check `MLIACTV` as described in the *ProDOS 8 Technical Reference Manual*.

- If an Escape key is detected in the keyboard buffer on an Apple IIc, it is removed. This is friendly to the Apple IIc Plus, the ROM of which does not remove the Escape key it uses to detect that the system should be booted at normal speed.

ProDOS 8 1.6

- Set up a parallel pointer to correct a PFI misinterpretation of an internal MLI pointer.

ProDOS 8 1.7

- Made a change to ensure that ProDOS 8 counts the volume's bitmap before incrementing the number of free blocks. This fixed a bug where an uninitialized location was being incremented and decremented, incorrectly reporting a `Disk Full` error where none should have occurred.
- Changed the handling of slot 3 ROM space to that described in ProDOS 8 Technical Note #15, How ProDOS 8 Treats Slot 3.
- Changed code to permit the invisible bit of the access byte (bit 2) to be set by applications.

ProDOS 8 1.8

- Fixed a bug introduced in V1.3. If an error occurs while calling `DESTROY` on a file, the file is not deleted but the index blocks are not swapped back to normal position. If a subsequent `DESTROY` of the same file succeeds, the volume's integrity is destroyed. Now ProDOS 8 marks the file as deleted, even if an error occurs, so any other errors do not cause a subsequent MLI call to trash the volume. Note that "undelete" utilities attempting to undelete such a file (one in which an error occurred during the `DESTROY`) may **trash** the volume.
- Fixed the `ONLINE` call to ignore the unused low nibble of the `unit_num` parameter when deciding how many bytes to zero in the application's buffer. This change fixes a bug which zeroed only the first 16 bytes of the caller's buffer before filling them if an `ONLINE` call was made with a `unit_num` of $0X, where X is non-zero.
- When loading on an Apple IIGS, ProDOS 8 now sets the video mode so the 80-column firmware is not active when the ProDOS 8 application gets control.
- Changed internal version checking between GS/OS and ProDOS 8. Note that GS/OS and ProDOS 8 are still tied to each other—versions that didn't come on the same disk can't be used together. The methods for checking versions were just altered.
- Made the backward compatibility check when opening subdirectories inactive. The test would always fail when opening a subdirectory with lowercase characters in the name (as assigned by the ProDOS FST under GS/OS), so the check was removed. Note that using earlier versions of ProDOS 8 with such disks causes errors when trying to access files with such directories in their pathnames.
- Expanded the ProDOS 8 loader code to provide for more room for future

compatibility.

- On a IIGS, installs a patch into the GS/OS stack-based call vector so that anyone calling GS/OS routines (like `QDStartUp` in ROM 03, for example) gets an appropriate error instead of performing a `JSL` into the stratosphere.

ProDOS 8 1.9

- New selector and dispatcher code was added for machines with 80 columns. The old code is still present for machines without 80-column capability.

- Fixed two bugs involved in booting into a ".SYSTEM" program larger than 38K. First, ProDOS 8 should be able to boot into a program as large as 39.75K, but was returning an error if the ".SYSTEM" program was larger than 38K. Second, when attempting to print the message "`*** SYSTEM PROGRAM TOO LARGE ***`", only one asterisk was printed. Both these bugs are fixed.

- No longer requires a ".SYSTEM" file when booting. If ProDOS 8 does not find a ".SYSTEM" file and the enhanced selector and dispatcher code is installed, ProDOS 8 executes a `QUIT` call.

- `KVERSION` is still $08.

Known Bugs in ProDOS 8 1.9

- ProDOS 8 doesn't handle seedling sparse files correctly. If you have a seedling file (only one block, less than 512 bytes of data) and use `SET_EOF` to grow the EOF without writing any data, the file becomes a sparse seedling file. The EOF indicates the file is a sapling or tree file, but in fact only one disk block is used. The storage type correctly indicates a seedling file.

  If you have such a file, and try to shrink it by setting the EOF to a value that is not an even multiple of a block boundary, ProDOS 8 gets confused and trashes some of the second half of the real data block.

- If you perform a `SET_PREFIX` to a slash (/) followed by a `SET_PREFIX` to a volume name not preceded by a slash, the prefix gets set to the first letter of the volume name.

Further Reference

- *ProDOS 8 Technical Reference Manual*
- *ProDOS 8 Update*
- *AppleShare Programmer's Guide to the Apple II*
- ProDOS 8 Technical Note #21, Identifying ProDOS Devices