



### ProDOS 8

#### #26: Polite Use of Auxiliary Memory

Written by: Matt “Missed Manners” Deatherage

January 1990

This Technical Note discusses the use of auxiliary memory, particularly the reserved areas, and this information supersedes the discussion in the *ProDOS 8 Technical Reference Manual*.

---

#### “I want to use auxiliary memory!”

Dear Missed Manners:

I’m having difficulty in a program I’m writing for 128K Apple II computers. My program is about to run out of memory. I have squeezed, packed and compressed this program until I can simply cajole no more room from it, and yet more room it needs. Apple has a large section of memory reserved, but my investigations reveal that this memory (in a language card, where it is doubly valuable since it stays put when main memory is swapped) seems to be unused. The *ProDOS 8 Technical Reference Manual* states unfailingly that the memory must not be used, but it seems to be wasting away! How can I politely use this valuable resource in my own application?

Gentle Developer:

Polite programming requires cooperation by both developers and system software, and it is the users who suffer when that cooperation is not maintained. Apple reserves memory for system software so that it can expand without breaking applications. Missed Manners hopes that he is not being too presumptuous by assuming that you would be appalled if Apple was required to expand ProDOS 8 and reclaim the memory from \$B000 through \$BFFF. He notes this situation would not be necessary if Apple were able to use memory it currently has reserved for such purposes.

However, if necessity requires more memory for your application, a polite inquiry to Apple may be sent. “Would it be possible for me to use some of Apple’s reserved memory in my application without compatibility problems?” would be a polite request, for example. Using the memory without asking or demanding action would not only be impolite, it would pose future problems for an application. Those who do not program politely will eventually regret such a decision.

## **Conflicts and Arbitration**

Some of the polite letters Apple has received on this subject point out that the built-in /RAM device uses almost all of the memory marked as “reserved” in the ProDOS 8 memory map. How can the system software expand into areas it’s already using?

It can't, of course...unless it already has and you don't know it. This is partially the case. On the Apple IIGS, memory can be obtained through the Memory Manager, so adding new components to the system software is relatively easy. If memory is available, it is allocated by the Memory Manager and used by the application. If memory is not available, the program trying to install the component in question is told and the component is not installed. (If a vital part of the system can't be installed, the boot process grinds to an unceremonious, but grammatically correct, halt.)

Since the 8-bit Apple II family has no memory manager, applications and system software must mutually (and politely) agree which areas of memory belong to whom. If the system software is broken into components, some memory will be reserved for components which are not present at a given time. This is largely the case with the auxiliary language card memory on the 128K Apple II.

The area from \$D100 through \$DFFF in bank 2 of the auxiliary language card is for the use of third-party RAM-based drivers, to be discussed in a future ProDOS 8 Technical Note. At least one version of Apple II SANE is configured to load at \$E000 in the auxiliary language card, which is perfectly acceptable since SANE is part of the system software (it just doesn't ship with the system).

Clearly, /RAM can't use this memory at the same time the system software does. This very dichotomy gives the Rule of Auxiliary Memory that simplifies this memory management.

**The Rule of Auxiliary Memory:** If /RAM is enabled, all auxiliary memory above location \$800 may be used by an application after first removing /RAM as discussed in the *ProDOS 8 Technical Reference Manual*. /RAM should be reinstalled upon completion.

If /RAM is not enabled, then auxiliary memory above \$800 may be used at the application programmer's discretion, but the areas marked as reserved must be respected.

System software use of this area should be denoted by the absence of /RAM. This means that if ProDOS 8 were to ever expand to run only on 128K machines and *require* auxiliary language card memory, that no /RAM device would be installed by default. Although this seems unlikely, it is nonetheless another indicator that your application should not depend on /RAM to operate.

Similarly, if /RAM is not present when your application is launched, you may **not** reenable it. If it is present, you may remove it to use the memory if you reinstall it when you're done.

Also note that auxiliary memory below \$800 that is not on the 80-column text screen is always reserved and may never be used by applications.

Applications which use reserved memory areas without observing this rule run the risk of storing data over third-party RAM-based drivers (rendering their software useless to peripherals that may require such drivers, like third-party networks, devices for the visually impaired, or closed-system hard disks) or future system software.

### **Further Reference**

---

- *ProDOS 8 Technical Reference Manual*