Apple II
Technical Notes

Developer Technical Support

# ProDOS 8
# #27: Hybrid Applications

Written by: Matt Deatherage March 1990

This Technical Note discusses considerations for "hybrid" applications, which use Apple IIGS-specific features from ProDOS 8.

## Why Use Hybrid Features?

There are many reasons not to write hybrid applications. If your target machine is the Apple IIGS, it's pretty silly to write a ProDOS 8-based application. You are limited to the slower I/O model of ProDOS 8, you cannot access foreign file systems or large CD-ROM volumes, you cannot reliably access the toolbox (patches to the toolbox are only loaded when GS/OS is booted, which forces you to require GS/OS to be booted), and you cannot work with desk accessories that do disk access (CDAs cannot reliably "save and restore" an area of bank zero to use for ProDOS 8 disk access because they don't know if an interrupt handling routine is located there).

However, applications targeted for all Apple II computers may reasonably wish to take advantage of IIGS features. For example, a word processor or telecommunications program may want to use extra IIGS memory. This Note is your spiritual guide to such features.

## Memory Management

Applications wishing to use extended (beyond the lower 128K) memory on the IIGS must, like all IIGS applications, get it from the Memory Manager. This is not a consideration for non-hybrid applications for two reasons. First, when GS/OS launches a ProDOS 8 program, it reserves all of the lower 128K memory for ProDOS 8, so no other component (tool, desk accessory, INIT) can accidentally use that memory. (In fact, if some of the memory is not

available, GS/OS refuses to launch ProDOS 8 at all.) Second, when ProDOS 8 is directly booted, none of the memory is allocated since these other components, which might be using the Memory Manager, aren't loaded either.

If your ProDOS 8 application was launched by GS/OS, all of the managed lower 128K has already been allocated for you by GS/OS. If you call `MMStartUp`, the user ID returned is one belonging to GS/OS. In such cases, the auxiliary field of the user ID is already being used by GS/OS and must **not** be altered by your application. You also must not call any Memory Manager routine which works on all handles of a given user ID, such as `DisposeAll` or `HUnlockAll`. You must manage all handles individually and not by user ID. You may, if you wish, call `GetNewID` to get a new user ID for use in a user ID-based memory management system. The ID should be of type $1000 (application).

You can tell whether your application was launched by GS/OS by checking `OS_BOOT`, the byte value at $E100BD. `OS_BOOT` is $00 when the boot OS was ProDOS 8, indicating that your application was not loaded by GS/OS. If this is the case and you want to use extended IIGS memory, you should call `GetNewID` to obtain a new application ID then use `NewHandle` to allocate four handles to hold the memory normally reserved for ProDOS 8 by GS/OS. You should obtain memory at $00/0800 (size $B800), $01/0800 (size $B800), $E0/2000 (size $4000) and $E1/2000 (size $8000). You may then use `MMStartUp` to register yourself with the Memory Manager; `MMStartUp` fails if it's being called from an unallocated memory block, so you must allocate the memory your application occupies first.

## Further Reference

- Apple IIGS Technical Note #17, Application Startup and the MMStartUp User ID