Apple II
File Type Notes

Developer Technical Support

File Type: $B6 (182)
Auxiliary Type: All

Full Name: ProDOS 16 or GS/OS Permanent Initialization File
Short Name: Permanent initialization file

Modified by: Matt Deatherage May 1992
Written by: Matt Deatherage September 1990

Files of this type contain initialization code that does not get unloaded.
Changes since September 1990: Added new information pertaining to System Software 6.0 and answered some commonly asked questions.

Files of type $B6 contain permanent initialization code in OMF format. Such files are often referred to as "inits." They are loaded by GS/OS at boot time and are never unloaded. The auxiliary type is **reserved** except for bit 15—if bit 15 is set, the initialization file is not loaded at boot time.

The structure of an init is similar to that of an application. The first byte of the loaded code image (inits are load files) is the entry point, and the init must end with an `RTL` instruction. When GS/OS transfers control to a permanent initialization file, the processor is in 16-bit native mode. The `A` register contains the init's user ID, `D` points to the bottom of a 4K stack and direct-page area and `S` points to near the top of that area. (If the init has an OMF stack and direct page segment linked in, the `D` and `S` registers point to it instead.) The data bank register is not defined; you should save it, set it and restore it if you use absolute addressing.

While all tools are available to be started, that doesn't mean tools should necessarily be started. Inits can be loaded after boot time (such as with IR 2.0, DTS Sample Code for an Apple IIGS Finder Extension), and blindly attempting to start and shut down tools without first checking their status can be disastrous in such instances. In particular, inits should never call `TLStartUp` or `TLShutDown`, and should check for the presence of other tools through each tool's status function before starting it up.

If you're considering starting a tool after init time and **leaving** it started (which is only possible when your code gets control after init time), you must do two things:

1. Apply for your Certificate of IIGS Deityship at the Matt & Dave Ministry of Bits. Do not continue until you recieve your certificate.
2. Read Apple IIGS Technical Note #53, "NDAs and Tools," and do what it says.

Inits that need to tell the difference between boot time and later loading times (for example, a RAM disk restoration init) can check the result of the GS/OS call `GetName`—if there is no name, the system's currently being started up.

Permanent inits are called at boot time and left in memory until the system is shut down. However, GS/OS does not call them again (even on a return from ProDOS 8). If your permanent init wants to periodically get control, it can use features like heartbeat tasks (installed with `SetHeartBeat` and `IntSource`), GS/OS notification procedures (`AddNotifyProc`), inter-process communication features (`AcceptRequests`) or Run Queue tasks (`AddToRunQ`).

Your permanent init can tell GS/OS it should be unloaded after execution. Above GS/OS's `RTL` address on the stack is a **Word** value of $0000. If your init sets bit zero of this word (`LDA #1`, `STA 4,S` assuming you haven't pushed anything on the stack), GS/OS unloads your init when you return control, treating it as if it were a temporary init file. This can be useful for inits that operate with certain hardware—if the hardware isn't present, the init can go away.

Warning: This **Word** space is not available to permanent initialization files that execute from a user's folder on an AppleShare file server at boot time unless you're using System Software 6.0 or later.

Further Reference

- *GS/OS Reference*
- File Type Note for File Type $B7, Temporary Initialization Files