

# Apple II File Type Notes

---



## Developer Technical Support

### **File Type:**           **\$5A (90)**

Full Name:   Application Configuration file  
Short Name:  Configuration file

Written by:   Matt Deatherage

May 1992

Files of this type and auxiliary type contain configuration or preferences.

---

Files of type \$5A contain configuration information or preferences for Apple II software. The original name for this file type is “Configuration,” but it’s friendlier to call such application-specific settings “preferences.” It’s easier for most people to understand—most people don’t “configure” things often, but everyone has their own preferences.

Any program can have preferences, although it’s most common for applications. However, inits, DAs and other system components can have preferences as well—see the DTS Sample Code “IR 2.0.1” for an example of an init with a preferences file.

Apple strongly recommends requesting a preferences auxiliary type if you create preferences files, so that other programs may identify your files (even if they don’t operate on them). However, if your files are named something very self-explanatory (such as the Sound Control Panel’s “Sound.Settings” file), you may use auxiliary type \$0000 **provided** you always identify your files by file name.

Preferences files belong in the same directory as the program that owns them unless the program is on a network file server, in which case the preferences should go in the user folder. If you don’t use the user folder, each user has to have write permission to the application folder (not always practical), and even so all users would have to use the same preferences. For GS/OS applications, proper preference management is easy—using the “@” prefix puts your files exactly where they should go. It’s a little trickier for non-applications, but the AppleShare-specific call `GetUserPath` makes it not too difficult. The previously-mentioned Sample Code (IR 2.0) shows how an init written in 65816 assembly language can place preference files properly.

Even if you do this, for registration and future identification purposes, we recommend you get an auxiliary type assignment. If you intend to identify your files by name and give them a self-explanatory name, you may request with your assignment that no custom File Type Descriptor string be included with the system software, which will save disk space and memory.

Note that using auxiliary type \$0000 when identifying preferences by name is an exception to the normal rule—in other file types, you may **not** use auxiliary type \$0000.

---

