

Apple II Technical Notes



Developer Technical Support

ProDOS 8

#19: File Auxiliary Type Assignment

Revised by: Matt Deatherage
November 1988

Written by: Matt Deatherage May
1988

This Technical Note describes file auxiliary type assignments.

The information in a ProDOS file auxiliary type field depends upon its primary file type. For example, the auxiliary type field for a text file (TXT, \$04) is defined as the record length of the file if it is a random-access file, or zero if it is a sequential file. The auxiliary type field for an AppleWorks™ file contains information about the case of letters in the filename (see Apple II File Type Notes, File Types \$19, \$1A, and \$1B). The auxiliary type field for a binary file (BIN, \$06) contains the loading address of the file, if one exists.

Auxiliary types are now used to extend the limit of 256 file types in ProDOS. Specific auxiliary types can be assigned to generic application file types. For example, if you need a file type for your word-processing program, Apple might assign you an auxiliary type for the generic file type of Apple II word processor file, if it is appropriate.

An application can determine if a given file belongs to it by checking the file type and the auxiliary type in the directory entry. Other programming considerations include the following:

1. If your program displays auxiliary type information, it should include all auxiliary types, not just selected ones. Try to display the auxiliary type information stored in the directory entry, just as you would display hex codes for file types for which you do not have a more descriptive message to display.
2. Programs should **not** store information in an undefined auxiliary type field. Storing the record length in a text file is fine, and it is even encouraged, but storing the number of words in a text file in that text file's auxiliary type field might cause problems for those programs which expect to find a record length there. Similarly, storing data in the auxiliary type field will cause problems if your data matches an auxiliary type which is assigned. To avoid these problems, only store defined items in a file's auxiliary type field. If you do not know of a definition for a particular file type's associated auxiliary type, do not store anything in its field.

To request a file type and auxiliary type, please send Apple II Developer Technical Support a description of your proposed file format, along with a justification for not using existing file and auxiliary types. We will publish this information publicly, unless you specifically prohibit it, since we feel doing so enables the exchange of data for those applications who choose to support other file formats.

Further Reference

- *ProDOS 8 Technical Reference Manual*
- *ProDOS 16 Technical Reference*