



### #18: StdFile

Written by: Keith Rollin

Versions:	1.00	April 1989
Components:	StdFile.c	April 1, 1989
	StdFile.p	April 1, 1989
	StdFile.h	April 1, 1989
	StdFile.r	April 1, 1989
	StdFile.rsrc	April 1, 1989
	CStdFile.make	April 1, 1989
	PStdFile.make	April 1, 1989

StdFile attempts to demonstrate the following techniques:

- Normal use of `SFGetFile` and `SFPutFile`.
- Normal use of `SFPGetFile` and `SFPputFile`, which includes the use of custom dialogs and handling extra items through the implementation of a `DlgHook`.
  - First time initialization.
  - Extra simple buttons (Quit, Directory, ThisDir).
  - Radio buttons (file format, types of files to show).
  - Aliasing—click on some buttons to click on other buttons.
  - Regenerating the list of files displayed.
- Creating a full pathname from a reply record (using working directories or `DirID`)
- Selecting a directory (*à la* MPWs “`GetFileName -d`”)
- Simple file filter (checks file type).
- Complex file filter (looking inside the file).
- Adding and deleting List Manager lists and extra List Manager lists. This is shown for both `SFGetFile` and `SFPutFile`.
- Select multiple files using one of two methods.
  - Replace StdFile’s list with one of your own.
  - Add a second list to the dialog box. This method is not shown explicitly, but rather, I show how to install and dispose of the actual list item. Inserting filenames into the list is left as an exercise to you, the programmer.
- Setting the starting directory or volume.
- Describe pending update event clogging.

**Note:** This application assumes the existence of HFS. It makes HFS calls and accesses HFS data structures without first checking to see if HFS exists on this machine.

In some cases, you will see me make use of a peculiar Pascal syntax: `IF <expr> THEN;`. This is intentional, as it gets the Pascal compiler to discard function results in which I have no interest.