

The /M flag is used to set the prompt message. As of OS X 10.11, Apple no longer shows the prompt message in the Save File dialog. It continues to work on Windows.

The /F flag is used to control the file filter which determines what kinds of files the user can create. This is explained further under **Save File Dialog File Filters**.

### Save File Dialog File Filters

The Save File dialog includes a file filter menu that allows the user to choose the type of file to be saved. By default this menus contain "Plain Text File" and, on Windows only, "All Files". You can use the /T and /F flags to override the default filter behavior.

The /T and /F flags work as explained under Open File Dialog File Filters. Using the /F flag for a Save File dialog, you would typically specify just one filter plus All Files, like this:

```
String fileFilters = "Data File (*.dat):.dat;"
```

```
fileFilters += "All Files.*;"
```

```
Open /F=fileFilters . . .
```

The file filter chosen in the Save File dialog determines the extension for the file being saved. For example, if the "Plain Text Files" filter is selected, the ".txt" extension is added if you don't explicitly enter it in the File Name edit box. However if you select the "All Files" filter then no extension is automatically added and the final file name is whatever you enter in the File Name edit box. You should include the "All Files" filter if you want the user to be able to specify a file name with any extension. If you want to force the file name extension to an extension of your choice rather than the user's, omit the "All Files" filter.

## Using Open in a Utility Routine

To be as general and useful as possible, a utility routine that acts on a file should have a `pathName` parameter and a `fileName` parameter, like this:

```
Function ShowFileInfo(pathName, fileName)
    String pathName    // Name of symbolic path or "" for dialog.
    String fileName    // File name or "" for dialog.

    <Show file info here>
End
```

This provides flexibility to the calling function. The caller can supply a valid symbolic path name and a simple leaf name in `fileName`, a valid symbolic path name and a partial path in `fileName`, or a full path in `fileName` in which case `pathName` is irrelevant.

If `pathName` and `fileName` fully specify the file of interest, you want to just open the file and perform the requested action. However, if `pathName` and `fileName` do not fully specify the file of interest, you want to display an Open File dialog so the user can choose the file. This is accomplished by using the **Open** operation's /D=2 flag.

With /D=2, if `pathName` and `fileName` fully specify the file, the Open operation merely sets the `S_fileName` output string variable to the full path to the file. If `pathName` and `fileName` do not fully specify the file, Open displays an Open File dialog and then sets the `S_fileName` output string variable to the full path to the file. If the user cancels the Open File dialog, Open sets `S_fileName` to "". In all cases, Open/D=2 just sets `S_fileName` and does not actually open the file.

If `pathName` and `fileName` specify an alias (*Macintosh*) or shortcut (*Windows*), Open/D=2 returns the file referenced by the alias or shortcut.

Here is how you would use Open /D=2.