

**#80: Standard File Tips**

See also:           The Standard File Package

Written by:         Jim Friedlander

June 7, 1986

Updated:

March 1, 1988

---

**SFSaveDisk and CurDirStore**

Low-memory location \$214 (SFSaveDisk—a word) contains  $-1 * \text{vRefNum}$  of the volume that SF is displaying (MFS and HFS). It never contains  $-1 * \text{WDRefNum}$ .

Low-memory location \$398 (CurDirStore—a long word) contains the `dirID` of the directory that SF is displaying (HFS only).

This information can be particularly useful at hook time, when the `vRefNum` field of the reply record has not yet been filled in. **Note:** `reply.fName` is filled in correctly at hook time if a file has been selected. If a directory has been selected, `reply.fType` is non-zero (it contains the `dirID` of the selected directory). If neither a file nor a directory is selected, both `reply.fName[0]` and `reply.fType` are 0.

**Setting Standard File's default volume and directory**

If you want `SFGetFile` or `SFPutFile` to display a certain volume when it draws its dialog, you can put  $-1 * \text{vRefNum}$  of the volume you wish it to display into the low-memory global `SFSaveDisk` (a word at \$214).

In Pascal, you would use something like:

```
...
TYPE
  WordPtr = ^INTEGER;           {pointer to a two-byte location}
CONST
  SFSaveDisk = $214;           {location of low-memory global}
VAR
  SFSaveVRef : WordPtr;
  myVRef      : INTEGER;
BEGIN
  ...
  {myVRef gets assigned here}
  ...
  SFSaveVRef := WordPtr(SFSaveDisk); {point to SFSaveDisk}
  SFSaveVRef^:= -1 * myVRef;         {"stuff" the value in}
  SFGetFile(...
```

In C you would use something like this (where a variable of type "short" occupies 2 bytes):

```
#define SFSaveDisk (*(short *)0x214)

short myVRef;
...
/* myVRef gets assigned here */
...
SFSaveDisk = -1 * myVRef; /* "stuff" the value in */
SFGetFile(...
```

If you are running HFS and would like to have Standard File display a particular directory as well as a particular volume, you can't just put a `WRefNum` into `SFSaveDisk`. If you do put a `WRefNum` into `SFSaveDisk`, Standard File will display the root directory of the default volume. Instead, you must put `-1 * the vRefNum` into `SFSaveDisk` (see above) and put the `dirID` of the directory that you wish to have displayed in `CurDirStore`. If you put an invalid `dirID` into `CurDirStore`, Standard File will display the root level of the volume referred to by `SFSaveDisk`. To change `CurDirStore` you can use a technique similar to the above, but remember that `CurDirStore` is a four-byte value. If your application is running under MFS, Standard File ignores `CurDirStore`, so you can use the same code regardless of file system.