HFS Notes Discussion of HFS Topics

This information is from Apple Technical Note #77.

What the HFS numbers mean:

- A Drive number is a small positive word (e.g. 3).
- A VRefNum (as opposed to a WDRefNum) is a small negative word (e.g. 0xFFFE).
- A WDRefNum is a large negative word (e.g. 0x8033).
- A DirID is a long word (e.g. 38). The root directory of an HFS volume always has a dirID of 2.

Most applications don't need to open working directories since **SFGetFile** returns a WDRefnum. Apple used to recommend that if you opened a working directory (WD), it should be created with an ioWDProcID of 'ERIK' and it would be deallocated by the Finder. However, Apple Technote #190 says that whenever you open a working directory with **PBOpenWD**, you should pass your application's signature as the ioWDProcID and close the working directory as soon as possible with **PBCloseWD**. You should only close working directories that you open, not those returned by Standard File or **SysEnvirons**. It is best to keep working directories open for the minimum time necessary, and to avoid using them when possible. Note that working directories are implemented mostly for the benefit of old (pre-HFS) applications, and rarely need to be used.

One more thing about WD's is that there are no WDRefNums that refer to the root. The root directory is always referred to by a vRefNum.

All of the HFS 'H' calls can be made without regard to the file system as long as you pass in a pointer to an HFS parameter block. **PBHGetVol**, **PBHSetVol** (don't use **PBHSetVol!**), **PBHOpen**, **PBHOpenRF**, **PBHCreate**, **PBHDelete**, **PBHGetFInfo**, **PBHSetFInfo**, **PBHSetFLock**, **PBHRstFLock** and **PBHRename** differ from their MFS counterparts only in that a dirID can be passed in at offset 0x30. The only difference between 'H' and MFS calls is that bit 9 of the trap word is set, which tells HFS to use a larger parameter block. MFS ignores this bit.

HFS-specific calls can only be made if HFS is available. The calls **PBGetCatInfo**, **PBSetCatInfo**, **PBOpenWD**, **PBCloseWD**, **PBGetFCBInfo**, **PBGetWDInfo**, **PBCatMove**, **PBDirCreate**,.and **PBSetVInfo** have no MFS equivalent. If any of these calls are made when running MFS, a system error will be generated. If **PBGetCatInfo** or **PBSetCatInfo** are called under MFS, they are translated into **PBGetFInfo** and **PBSetFInfo** calls.

If HFS is running, a call to **GetVol** (before you have made any calls to **SetVol**) will return the WDRefNum of your application's parent directory in the vRefNum parameter. If your application was launched by the user clicking on one or more documents, the WDRefNums of those documents' parent directories are available in the vRefNum field of the AppFile record returned from **GetAppFiles**.

If HFS is running, the PMSP (Poor Man's Search Path) is used for any file system call that can return a file-not-found error, such as **PBOpen**, **PBClose**, **PBDelete**, **PBGetCatInfo**, etc. It is not used for indexed calls (where ioFDirIndex is positive) or when a file is created (**PBCreate**) or

when a file is being moved between directories (**PBCatMove**). The PMSP is also not used when a non-zero dirID is specified.

If you want to get information about a volume, but you only have its name and aren't even sure that it's the proper one, you should set MyHPB.ioVRefNum to -32768 (0x8000). No vRefNum or WDRefNum can be equal to 0x8000. By doing this, you are forcing **PBHGetVInfo** to use the volume name and, if that name is invalid, to return a nsvErr(-35, "No such volume"). If you pass any other value in ioVRefNum (including zero), and the volume name turns out to be invalid (maybe you forgot the colon part of the name, for example), **PBHGetVInfo** returns information about the default volume, and an error code of 0 (noErr).