PBDelete Delete closed file or empty directory

#include <Files.h> **File Manager (PBxxx)**

```
OSErr       PBDelete(pb, async);
ParmBlkPtr  pb ;        address of an 80-byte FileParam structure
Boolean     async ;     0=await completion; 1=immediate return
            returns     Error Code; 0=no error
```

PBDelete deletes both forks of a file, freeing up all its storage. You can use this call to delete an empty directory.

pb is the address of an 80-byte FileParam structure (or a fileParam member of an ParamBlockRec union). The relevant fields are as follows:

Out-In	Name	Type	Size	Offset	Description
->	ioCompletion	ProcPtr	4	12	Completion routine address (if *async* =TRUE)
->	ioNamePtr	StringPtr	4	18	Address of full or partial path/filename
->	ioVRefNum	short	2	22	Volume, drive, or working directory reference
->	ioFVersNum	SignedByte	1	26	Version (always 0 on HFS)
<-	ioResult	OSErr	2	16	Error Code (0=no error, 1=not done yet)

async is a Boolean value. Use FALSE for normal (synchronous) operation or TRUE to enqueue the request and resume control immediately. See Async I/O.

Returns: an operating system Error Code. It will be one of:

noErr	(0)	No error
bdNamErr	(-37)	Bad name
extFSErr	(-58)	External file system
fBsyErr	(-47)	File is busy, dir not empty, working dir still open
fLckdErr	(-45)	File is locked
fnfErr	(-43)	File not found
ioErr	(-36)	I/O error
nsvErr	(-35)	No such volume
vLckdErr	(-46)	Volume is locked
wPrErr	(-44)	Diskette is write-protected

Notes: The file's directory entry is erased from the disk and its storage (both forks) is released. This does NOT place the file in some unspecified, recoverable "trash can"; however, the file's data does remain intact until it is overwritten by another file. Given a robust file-recovery utility, you can still recover the file after deletion.

All access paths to the file must be closed (see **PBClose**). To remove a directory, it must be empty and all working directory control blocks for that directory must be closed (see **PBCloseWD**). Remember that Standard File opens working directories and with Switcher or MultiFinder, several processes may have working directories opened. Use **PBGetWDInfo** to get a list of open working directories.

PBDelete will not delete a locked file or directory.

Example

```
#include <Files.h>

FileParam fpb;
Str255        fileName="\pHD 20:Ltrs:1988:Jones";
OSErr      rc;

fpb.ioNamePtr = fileName;
fpb.ioVRefNum = 0;                  // ignored if name is valid
fpb.ioFVersNum = 0;                 // ignored with HFS
rc=PBDelete( &fpb, FALSE );
if ( rc ) { /* . . . handle the error . . . */ }
```