PBHDelete Delete closed file or empty directory (HFS only)

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

OSErr	**PBHDelete(***pb*, *async* **);**	
HParmBlkPtr	*pb* ;	address of an 80-byte HFileParam structure
Boolean	*async* ;	0=await completion; 1=immediate return
	returns	Error Code; 0=no error

PBHDelete deletes both forks of a file, freeing up all its storage, or deletes an empty directory. This allows you to specify a directory using a "hard" directory ID. If a file ID reference exists for the file being deleted, the file ID reference is also deleted.

pb is the address of an 80-byte HFileParam structure. 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
–>	ioDirID	long	4	48	Directory ID (0=use name and/or refNum)
<-	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
dirNFErr	(-120)	Directory not found or incomplete pathname
extFSErr	(-58)	External file system
fBsyErr	(-47)	File is busy, directory not empty, or working dir 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: **PBHDelete** works like **PBDelete**, except that it provides a way to supply a directory ID and file version numbers are ignored. See that topic for related details.

If you use a directory ID in ioDirID, it will override any volume or working-directory number in ioVRefNum. If you wish to use a working directory reference (as obtained via Standard File), put it in ioVRefNum and use ioDirID=0.