HDelete Delete an unopened file or empty directory

#include <Files.h> **File Manager**

```
OSErr    HDelete( vRefNum, dirID, fileName );
short    vRefNum ;    volume or working directory reference
long     dirID;       ID of directory where the file resides
Str255   fileName ;   address of length-prefixed full or partial name
         returns      Error Code; 0=no error
```

HDelete deletes both forks of a file. The file must not be open. This function can also be used to delete an empty directory.

vRefNum is the reference number of the volume or working directory containing the file or directory *fileName*. Use 0 to specify the default volume.

dirID is the ID of the directory that contains the file to be deleted.

fileName is the address of a length-prefixed, pascal-style string containing the name of the file to be deleted. It may be a partial or full pathname, depending upon the value of *vRefNum* .

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
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: If the file to delete is currently open (or if the directory to delete contains any files or directories) **HDelete** will fail, returning an Error Code.

Use **PBDelete** if you need to specify a file version number. If you want to clear out an entire directory, use **PBGetCatInfo** to index through all entries in the directory.

Note that this is a permanent deletion, and not a retrievable transfer to a friendly "trash can". However, a good disk utility package can recover the file data as long as no new data is written over it.