PBExchangeFiles Swap the data stored in two files

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

```
OSErr       PBExchangeFiles(pb, async);
HParmBlkPtr pb ;        pointer to a file ID parameter block
Boolean     async ;     0=await completion; 1=immediate return
            returns     Error Code; 0=no error
```

Use the **PBExchangeFiles** function to swap the data stored in two files on the same volume.

pb is a pointer to a FIDParam structure.The relevant fields are as follows:

Out-In	Name	Type	Size	Offset	Description
→	ioCompletion	ProcPtr	4	12	pointer to completion routine
←	ioResult	short	2	16	result code
→	ioNamePtr	long	4	18	pointer to first filename
→	ioVRefNum	short	2	22	volume specification (volume reference number, working directory reference number, drive number, or 0 for default volume)
→	ioMisc	Ptr	4	28	pointer to second filename
→	ioDestDirID	long	4	36	second parent directory ID
→	ioSrcDirID	long	4	48	first parent directory ID

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
nsvErr	(-35)	Volume not found
ioErr	(-36)	I/O error
fnfErr	(-43)	File not found
fLckdErr	(-45)	File locked
volOfflinErr	(-53)	Volume is off line
extFSErr	(-58)	External file system
wrgVolTypeErr	(-123)	Not an HFS volume
diffVolErr	(-1303)	Files on different volumes (**FSSpec** is still valid)

Notes: **PBExchangeFiles** swaps the data in two files by changing the information in the volume catalog and, if the files are open, in the file control blocks. See **Using FSSpec Records** under the section entitled **Using the File Manager** for an illustration of how **PBExchangeFiles** changes the catalog entries and file control blocks.

You should use **PBExchangeFiles** or **FSpExchangeFiles** to preserve the file ID when updating an existing file, in case the file is being tracked through its file ID.

Typically, you use **PBExchangeFiles** after creating a new file during a safe save. You identify the names and parent directory IDs of the two files to be exchanged in the fields ioNamePtr, ioDestNamePtr, ioSrcDirID, and ioDestDirID. **PBExchangeFiles** changes the fields in the catalog entries that record the location of the data and the modification dates. It swaps both the data forks and the resource forks.

PBExchangeFiles works on either open or closed files. If either file is open, **PBExchangeFiles** updates any file control blocks associated with the file. Exchanging the contents of two files requires essentially the same access as opening both files for writing.

PBExchangeFiles does not require that file IDs exist for the files being exchanged.