PBSetFPos Set position of an open file's file mark

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

```
OSErr       PBSetFPos(pb, async);
ParmBlkPtr  pb ;        address of a 50-byte IOParam structure
Boolean     async ;     0=await completion; 1=immediate return
            returns     Error Code; 0=no error
```

PBSetFPos selects a new position for the file mark (the place where the next read or write will start) of an open file.

pb is the address of a 50-byte IOParam structure. The relevant fields are:

Out-In	Name	Type	Size	Offset	Description
->	ioCompletion	ProcPtr	4	12	Completion routine address (if *async* =TRUE)
->	ioRefNum	short	2	24	File reference number
->	ioPosMode	short	2	44	Positioning Mode (1=absolute, 2=from EOF, et.al)
<->	ioPosOffset	long	4	46	Entry: Position delta (bytes from start,EOF,...) Return: New value of mark (from start of file)
<-	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
eofErr	(-39)	End of file
extFSErr	(-58)	External file system
fnOpnErr	(-38)	File not open
ioErr	(-36)	I/O error
posErr	(-40)	Can't position to before start of file
rfNumErr	(-51)	Bad ioRefNum

Notes: Since you can set ioPosOffset when you call **PBRead** and **PBWrite**, **PBSetFPos** is not needed often.

In random-access database operations, watch for the eofErr return value. You may need to enlarge the file via **PBSetEOF** or **PBAllocate**.