PBAllocate Increase file allocation; logical EOF unchanged

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

OSErr	**PBAllocate(***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

PBAllocate increases the physical size of an open file, adding one or more allocation blocks to the end of the file. The file must be opened with a read/write permission level.

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

Out-In	Name	Type	Size	Offset	Description
–>	ioCompletion	ProcPtr	4	12	Completion routine address (if *async* =TRUE)
–>	ioRefNum	short	2	24	File reference number
–>	ioReqCount	long	4	36	Desired disk space to add to the file, in bytes
<-	ioResult	OSErr	2	16	Error Code (0=no error, 1=not done yet)
<-	ioActCount	long	4	40	Actual amount of space added, in bytes

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
fLckdErr	(-45)	File is locked
fnOpnErr	(-38)	File not open
ioErr	(-36)	I/O error
rfNumErr	(-51)	Bad ioRefNum
vLckdErr	(-46)	Volume is locked
wPrErr	(-44)	Diskette is write-protected
wrPermErr	(-61)	Write permissions error

Notes: **PBAllocate** expects the reference number of an open file in ioRefNum (it must be unlocked, opened with write permission, and it must exist on an unlocked volume).

ioReqCount bytes are added to the current physical EOF and the result is rounded up to the next higher allocation block size. For instance, if ioReqCount=1, the physical EOF will increase by 512 bytes and ioActCount will return containing 512.

If there isn't enough room on the disk to satisfy the request, then all available storage is added to the file and dskFulErr is returned. Compare this to **PBAllocContig** and **PBSetEOF** (which do NOT perform such absurd partial allocations).

The **PBAllocContig** function may be preferable. It attempts to find a contiguous block of disk space of the requested size (contiguous blocks make for faster file access).