PBAllocContig Increase physical EOF as a contiguous block

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

OSErr	**PBAllocContig(***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

PBAllocContig locates a contiguous series of disk blocks and adds that storage to the physical EOF of an open 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
dskFulErr	(-34)	Disk full
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: **PBAllocContig** works like **PBAllocate** (adding blocks of disk storage to the end of an open file). It differs in that it attempts to locate sufficient empty space as a contiguous series of blocks. If it can't find such a block, the function fails, without taking action, returning dskFulErr.

A typical way to optimize disk access for sequential files (e.g., wordprocessing documents), is to use **PBSetEOF** to truncate a file to 0-length, followed by **PBAllocContig** to allocate a contiguous storage area (if it fails, try **PBAllocate**). Then use **PBWrite** to write data to the file (fastest operation is to write in 512-byte chunks). Use **PBSetEOF**, if needed, to release any unused blocks at the end of the allocation.