PBUnlockRange Restore global access to a portion of a shared file

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

```
OSErr       PBUnlockRange(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
```

After locking a portion of a file via **PBLockRange**, perform any required updating of the region, and then use **PBUnlockRange** to release the lock.

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

Out-In	Name	Type	Size	Offset	Description
->	ioRefNum	short	2	24	File reference number
->	ioReqCount	long	4	36	Size of region to lock, in bytes
->	ioPosMode	short	2	44	Positioning Mode (1=absolute, 2=from EOF, et.al)
->	ioPosOffset	long	4	46	Positioning delta (bytes from start, EOF, et.al)
->	ioCompletion	ProcPtr	4	12	Completion routine address (if *async* =TRUE)
<-	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
paramErr	(-50)	Range size is less than 0
rfNumErr	(-51)	Bad ioRefNum value

Notes: See **PBLockRange** for an example of usage.

PBUnlockRange is not supported by the 64K ROM File Manager.

Use **PBSetFLock**...**PBRstFLock** to lock/unlock the whole file or use **PBSetVInfo** to lock/unlock an entire volume.