

PBRstFLock Unlock a file (allow write access)

#include <Files.h>

File Manager (PBxxx)

OSErr **PBRstFLock**(*pb*, *async*);
ParmBlkPtr *pb* ; address of an 80-byte FileParam structure
Boolean *async* ; 0=await completion; 1=immediate return
returns Error Code; 0=no error

PBRstFLock unlocks a file. It undoes the effect of **PBSetFLock**, allowing the file to be deleted or modified.

pb is the address of an 80-byte FileParam structure. The relevant fields are as follows:

<u>Out-In Name</u>	<u>Type</u>	<u>Size</u>	<u>Offset</u>	<u>Description</u>
-> ioCompletion	<u>ProcPtr</u>	4	12	Completion routine address (if <i>async</i> =TRUE)
-> ioNamePtr	<u>StringPtr</u>	4	18	Address of full or partial path/filename
-> ioVRefNum	<u>short</u>	2	22	Volume, drive, or directory reference
-> ioFVersNum	<u>SignedByte</u>	1	26	Version (usually 0, always 0 for HFS)
<- ioResult	<u>OSErr</u>	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
extFSErr	(-58)	External file system
fnfErr	(-43)	File not found
ioErr	(-36)	I/O error
nsvErr	(-35)	No such volume
vLckdErr	(-46)	Volume is locked
wPrErr	(-44)	Diskette is write-protected

Notes: See **PBSetFLock** for related information.

Be sure to call **PBFlushVol** to check that the change is written to the disk.