HRstFLock Unlock a file (allow changes, deletion, renaming, etc.)

#include <Files.h> **File Manager**

OSErr	**HRstFLock(***fileName, dirID, vRefNum* **);**	
short	*vRefNum* ;	volume or working directory reference
long	dirID;	directory ID
Str255	*fileName* ;	address of length-prefixed full or partial name
	returns	Error Code; 0=no error

HRstFLock unlocks a file; it undoes the effect of **SetFLock**. This allows programs to delete, rename, or write data to it.

vRefNum is the reference number of the volume or working directory that contains the file or directory *fileName*. Use 0 to specify the default volume.

dirID is the directory ID of the directory where the file resides.

fileName is the address of a length-prefixed, pascal-style string containing the name of the file to be unlocked. It may be a partial or full pathname, depending upon the value of *vRefNum* .

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: This clears the file's "lock" flag (as found in the ioFlAttrib field of the FileParam structure) and notifies the system of the change (Note: changing this bit directly, e.g., via **PBSetCatInfo**, may not be noticed by the Finder until the file's folder is closed and reopened or the system is restarted).

This does not affect currently-open access paths. Thus, if some other process has opened the file, locking it will not prevent the other program from continuing to modify it.

See **SetFLock** for related details. You can lock/unlock an entire volume via **PBSetVInfo** or lock a selected portion of an open file via **PBLockRange**. Use **PBGetFInfo** to see if a file is currently locked (ioFlAttrib bit 1 is set).