GetEOF Obtain the size of an open file (logical EOF)

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

```
OSErr   GetEOF(fRefNum, curEOF );
short   fRefNum ;     file reference as obtained via FSOpen
long    *curEOF ;     receives size of file, in bytes
        returns       Error Code; 0=no error
```

Use **GetEOF** to find the current size of a file (its logical end-of-file position).

fRefNum is the reference number of an open file. See **FSOpen** and **OpenRF**.

curEOF is the address of long integer. Upon return, it will contain the file position of the logical end-of-file; i.e., the size of the file, in bytes.

Returns: an operating system Error Code. It will be one of:

noErr	(0)	No error
extFSErr	(-58)	External file system
fnOpnErr	(-38)	File not open
ioErr	(-36)	I/O error
rfNumErr	(-51)	Bad *fRefNum*

Notes: Use **GetEOF** to learn the size of file. This and all high-level file operations refer to the logical end-of-file, as opposed to the physical EOF.

Note: The **physical** EOF is always greater than or equal to the **logical** EOF, is a multiple of the size of an allocation unit (usually 1K), and has no significance for most applications.

The following example opens a file, allocates a memory buffer to hold all of its data, and reads the data into the buffer. See **OpenRF** for an example program that copies the contents of both forks of one file to another and uses this function to learn the size of the file.

Example

```
#include <Files.h>
#include <Memory.h>

short       fRef, rc;
long        fileSize;
Handle      hData;                  /* handle to buffer to be allocated */

rc = FSOpen( "\pHardDisk:MyFile", 0, &fRef );
if ( rc ) { /* . . . handle the error . . . */ }

rc = GetEOF( fRef, &fileSize );             /* get file size */
if ( rc ) { /* . . . handle the error . . . */ }

hData = NewHandle( fileSize );              /* allocate enough RAM */
if ( hData == 0 ) { /* . . . handle the error . . . */ }

rc = FSRead( fRef, &fileSize, *hData );     /* read it in */
```

```
if ( rc ) { /* . . . handle the error . . . */ }

FSClose( fRef );
```