

PBHOpenRF

Open file resource fork (HFS only)

#include <Files.h>

File Manager (PBxxx)

OSErr **PBHOpenRF**(*pb*, *async*);
HParmBlkPtr *pb*; address of a HParmBlockRec union
Boolean *async*; 0=await completion; 1=immediate return
returns Error Code; 0=no error

PBHOpenRF opens the resource fork of a file, enabling I/O operations. As with **OpenRF**, this is normally used only in file copy operations.

pb is the address of a 122-byte HParmBlockRec union. This call uses members of two different structures. The following structures and fields are relevant:

<u>Out-In</u>	<u>Name</u>	<u>Type</u>	<u>Size</u>	<u>Offset</u>	<u>Structure</u>	<u>Description</u>
->	ioCompletion	<u>ProcPtr</u>	4	12	ioParam	Completion rtn address (used only if <i>async</i> =TRUE)
->	ioNamePtr	<u>StringPtr</u>	4	18	ioParam	Address of full or partial path/filename
->	ioVRefNum	<u>short</u>	2	22	ioParam	Volume, drive, or directory ref
->	ioPermsn	<u>SignedByte</u>	1	27	ioParam	File Permission (1=read, 2=write...)
->	ioMisc	<u>Ptr</u>	4	28	ioParam	Address of 522-byte buf (0=use vol buf)
->	ioDirID	<u>long</u>	4	48	fileParam	Directory ID (0=use ioVRefNum)
<-	ioResult	<u>OSErr</u>	2	16	ioParam	Error Code (0=no err, 1=not done yet)
<-	ioRefNum	<u>short</u>	2	24	ioParam	File reference number

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
bdNamErr	(-37)	Bad name
dirNFErr	(-120)	Directory not found or incomplete pathname
extFSErr	(-58)	External file system
fnfErr	(-43)	File not found
ioErr	(-36)	I/O error
nsvErr	(-35)	No such volume
opWrErr	(-49)	File already open for writing
permErr	(-54)	Attempt to open locked file for writing
tmfoErr	(-42)	Too many files open

Notes: Use **PBOpenRF** or **PBHOpenRF** or **OpenRF** if you wish to treat the resource fork of the file as if it were unformatted data. The only legitimate use for these calls is in file copying operations. See **OpenRF** for an example program which duplicates both forks of a file.

See **PBHOpen** for a discussion of how you need to use multiple structures to access all fields of the parameter block.