**HFileParam** structure

```
#include <Files.h>

typedef struct HFileParam {           Size  Offset  Description
  struct QElem * qLink;               4     0       Address of next queue element
                                                    (0=last)
  short          qType;               2     4       Always ioQType (2)
  short          ioTrap;              2     6       (used internally by File Manager)
  Ptr            ioCmdAddr;           4     8       (used internally by File Manager)
  ProcPtr        ioCompletion;        4     12      Completion routine address
                                                    (see Async I/O)
  OSErr          ioResult;            2     16      Error code (0=no error, 1=not done
                                                    yet, ...)
  StringPtr      ioNamePtr;           4     18      Addr of full or partial path/filename
  short          ioVRefNum;           2     22      Volume, drive, or directory
                                                    reference
  short          ioFRefNum;           2     24      File reference number
  char           ioFVersNum;          1     26      Version (use 0 for HFS)
  char           filler1;             1     27      (unused)
  short          ioFDirIndex;         2     28      Index
  char           ioFlAttrib;          1     30      File Attribute bits (locked,
                                                    directory, etc)
  char           ioFlVersNum;         1     31      File version (best to use 0)
  FInfo          ioFlFndrInfo;        16    32      File type, creator, flags, etc. (see
                                                    FInfo)
  long           ioDirID;             4     48      'Hard' Directory ID
  unsigned short ioFlStBlk;           2     52      First allocation block of data fork
  long           ioFlLgLen;           4     54      Logical end-of-file of data fork
  long           ioFlPyLen;           4     68      Physical end-of-file of data fork
  unsigned short ioFlRStBlk;          2     62      First allocation block of resource
                                                    fork
  long           ioFlRLgLen;          4     64      Logical end-of-file of resource fork
  long           ioFlRPyLen;          4     68      Physical end-of-file of resource
                                                    fork
  unsigned long ioFlCrDat;            4     72      Date/time of creation (seconds since
                                                    1/1/04)
  unsigned long ioFlMdDat;            4     76      Date/Time of last modification
} HFileParam;                         80
```

Notes: The HFileParam structure is used HFS-specific calls (**PBH**xxx) calls which typically operate on closed files:

| | | | |
|---|---|---|---|
| **PBDirCreate** | **PBHGetFInfo** | **PBHRename** | **PBHSetFLock** |
| **PBHCreate** | **PBHOpen** | **PBHRstFLock** | |
| **PBHDelete** | **PBHOpenRF** | **PBHSetFInfo** | |

It is identical to FileParam in length, but the ioDirID field at offset 48 has changed names and meanings. Use this to specify a 'Hard' directory ID if you happen to have one. Use 0 to use the normal volume or working directory reference number in ioVRefNum.

The most common way to use this structure is to allocate a union which is an aggregate and create and initialize a pointer to the desired data type.  See HParamBlockRec for an example.