GetDrvQHdr Obtain pointer to the drive queue header

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

QHdrPtr **GetDrvQHdr( );**
returns address of a 10-byte QHdr structure

GetDrvQHdr returns the address of the header of the standard Operating System queue used to maintain the linked-list of disk drive information records. There is one DrvQEl entry for each physical drive attached to the Mac.

Returns: a 32-bit QHdrPtr; the address of the 10-byte QHdr structure whose qLink field points to the first DrvQEl structure in the queue.

Notes: C programmers may prefer to get this address from the global variable DrvQHdr (at 0x0308).

You can use this function to obtain information about drives attached to the system; it may be the only way to get this collection of information. If you need to monkey with this queue, you can use **Enqueue** and **Dequeue**.

In addition to the information in the DrvQEl structure, there are four bytes of additional data that precede each element. The following describes these prefix bytes:

Offset	Description
-4	(bit 7 set) = disk is locked (write-protected)
-3	0 = no disk in drive 1 or 2 = disk is in drive 8 = non-ejectable disk FCh...FFh = disk was ejected within last 1.5 seconds 48h = non-ejectable disk, but driver expects a call
-2	(used internally during system startup)
-1	(bit 7 clear) = drive supports only single-sided media

The following example reads the elements of the drive queue and displays information about all drives attached to the system.

Example

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

QHdrPtr     qhp;                    /* address of a QHdr structure */
DrvQEl      *qep;
Byte        *bp;                    /* helps to decode prefix bytes */
long        totBlks;
Boolean     locked, oneSide, empty;

qhp= GetDrvQHdr();                  /* address of queue header */
qep = (DrvQEl *)qhp->qHead;         /* address of a queue element */

printf("Drv# FileSys Blocks locked 1-sided empty\n");
do {
    bp=(Byte *)qep;  bp -=4;        /* point to structure prefix bytes */
```

```
        if ( qep->qType == 0 )
            totBlks = qep->dQDrvSz;       /* get size (in logical blocks) */
        else
            totBlks = qep->dQDrvSz + (((long) qep->dQDrvSz2) << 16);

        locked = oneSide = empty = FALSE;         /* set Booleans from prefix flags */
        if ( (bp[0] & 0x80)==0x80 ) locked=TRUE;
        if ( (bp[3] & 0x80)==0 ) oneSide=TRUE;
        if ( bp[1]==0 ) empty=TRUE;

        printf(" %d  %5d %8ld    %s    %s     %s\n",
                qep->dQDrive,
                qep->dQFSID,
                totBlks,
                locked  ? "Yes" : "No ",
                oneSide ? "Yes" : "No ",
                empty   ? "Yes" : "No "
        );
        qep=(DrvQEl *)qep->qLink;                 /* point to next queue element */
    } while( qep != 0 );
```