DrvQEl structure

#include <Files.h>

typedef struct {

		Size	Offset	Description
struct QElem * qLink;		4	0	Address of next queue element (0=last)
short	qType;	2	4	0=dQDrvSz is drive size, 1=very large
short	dQDrive;	2	6	Drive number
short	dQRefNum;	2	8	Driver reference number
short	dQFSID;	2	10	File-system identifier (0=native, else=other)
short	dQDrvSz;	2	12	Blocks on disk (total bytes or low-word)
short	dQDrvSz2;	2	14	Hi word of disk block count (if qType=1)
} **DrvQEl**;		16		

typedef DrvQEl ***DrvQElPtr**;

Notes: The meaning of the qType field has been corrupted. If HFS is present, then this field indicates how to calculate the size of the drive. It is one of:

0 The drive contains dQDrvSz blocks (probably 512-bytes each)

1 The drive contains dQDrvSz +(dQDrvSz * 0x10000) blocks

3 MFS is probably present and dQDrvSz is the total blocks

Preceding this structure in memory is a 4-byte set of flags, defined as follows:

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

See **GetDrvQHdr** for an example of how to access these prefix bytes and how to calculate the true size of the drive.

The global variable DrvQHdr contains the queue header which begins the chain of drive queue elements. Or call **GetDrvQHdr** to obtain that address.

dQDrvSz2 is only used if qType is 1. In this case, dQDrvSz2 contains the high-order word of the size, and dQDrvSz contains the low-order word.