HVolumeParam structure

#include <Files.h>

typedef struct **HVolumeParam** {		Size	Offset	Description
ParamBlockHeader		24	0	common fields of ParamBlock types
long	filler2;	4	24	(reserved)
short	ioVolIndex;	2	28	(>0: index, <0: use name/num, 0: use num)
unsigned long	ioVCrDate;	4	30	Date/time volume created
unsigned long	ioVLsMod;	4	34	Date/time volume information was modified
short	ioVAtrb;	2	38	Volume Attributes
unsigned short	ioVNmFls;	2	40	Count of files in the root directory
short	ioVBitMap;	2	42	Sector of start of volume bit map
short	ioAllocPtr;	2	44	Block at which next new file will start
unsigned short	ioVNmAlBlks;	2	46	Total allocation blocks on the volume
long	ioVAlBlkSiz;	4	48	Size of an allocation block, in bytes
long	ioVClpSiz;	4	52	Default allocation clump size, in bytes
short	ioAlBlSt;	2	56	First sector represented in bit map (flat vols)
long	ioVNxtCNID;	4	58	Next serial number for new file or directory
unsigned short	ioVFrBlk;	2	62	Number of unused allocation blocks
unsigned short	ioVSigWord;	2	64	Volume signature: 0xD2D7=flat, 0x4244=HFS
short	ioVDrvInfo;	2	66	Drive number
short	ioVDRefNum;	2	68	Driver reference number
short	ioVFSID;	2	70	File system identifier (0=native; else=external)
unsigned long	ioVBkUp;	4	72	Date/Time of last backup
unsigned short	ioVSeqNum;	2	76	Sequence number (if this is a backup diskette)
long	ioVWrCnt;	4	78	Volume write count
long	ioVFilCnt;	4	82	Total number of files on the volume
long	ioVDirCnt;	4	86	Total number of directories on the volume
long	ioVFndrInfo[8];	32	90	Data used by Finder (mostly undocumented)
			90	(long) dirID of the 'blessed folder'
			94	(long) dirID of directory of startup application
			98	(long) dirID of first folder in Finder chain
			. . .	(other undocumented fields)
} **HVolumeParam**;		122		

Notes: The **HVolumeParam** structure is used specifically in only the **PBHGetVInfo** and **PBSetVInfo** functions which query or change information about HFS volumes.

Some fields match up with fields from the older VolumeParam structure and others are extracted from the VCB structure (volume data maintained in the volume control block queue). Other fields are added, removed, or names changed, apparently at random, to keep us guessing.

See **GetVCBQHdr**, **GetDrvQHdr**, **SysEnvirons**, and **FindFolder** for other ways to obtain information about volumes and disks.

The ioVAtrb field is a set of bit flags. See Volume Attributes.

The states of ioVDRefNum and ioVDrvInfo can be used to see if a volume is currently on-line, off-line, or ejected:

On-line:	ioVDRefNum < 0	and	ioVDrvInfo > 0
Off-line:	ioVDRefNum < 0	and	ioVDrvInfo = 0
Ejected:	ioVDRefNum = 0	and	ioVDrvInfo > 0

The 'blessed' folder for this volume, found at ioVFndrInfo[0], is the folder which contains the system file and the Finder. Use **SysEnvirons** or **FindFolder** to find the 'blessed' folder which contains the *currently open* system file.

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 examples.