PBHGetVInfo Get information about an HFS volume

#include <Files.h> **File Manager (PBxxx)**

OSErr	**PBHGetVInfo(***pb*, *async* **);**	
HParmBlkPtr	*pb* ;	address of a 122-byte HVolumeParam structure
Boolean	*async* ;	0=await completion; 1=immediate return
	returns	Error Code; 0=no error

PBHGetVInfo obtains a variety of information about a specified volume or directory, much as **PBGetVInfo**. It uses a longer, richer parameter block, including the time/date of the last backup and the total number of files and folders on the volume or in the specified directory.

pb is the address of a 122-byte HVolumeParam structure. The following fields are relevant:

Out-In	Name	Type	Size	Offset	Description
->	ioCompletion	ProcPtr	4	12	Completion routine address (if async =TRUE)
->	ioVolIndex	short	2	28	(>0=index, <0=use name/num, 0=use num)
<->	ioNamePtr	StringPtr	4	18	Entry: Address of full or partial pathname Return: receives volume name
<->	ioVRefNum	short	2	22	Volume, drive, or working directory reference
<-	ioResult	OSErr	2	16	Error Code (0=no error, 1=not done yet)
<-	ioVCrDate	long	4	30	Date/time volume created
<-	ioVLsMod	long	4	34	Date/time volume information was modified
<-	ioVAtrb	short	2	38	Volume attributes (bit 15=locked, etc.)
<-	ioVNmFls	short	2	40	Count of files in the root (or specified) directory
<-	ioVBitMap	short	2	42	First block of volume allocation bit map
<-	ioAllocPtr	short	2	44	Block at which next new file starts
<-	ioVNmAlBlks	short	2	46	Count of all allocation blocks in volume
<-	ioVAlBlkSiz	long	4	48	Allocation block size, in bytes
<-	ioVClpSiz	long	4	52	Default clump size (bytes to allocate)
<-	ioAlBlSt	short	2	56	First block in volume block map
<-	ioVNxtCNID	long	4	58	Next unused file number
<-	ioVFrBlk	short	2	62	Number of free allocation blocks
<-	ioVSigWord	short	2	64	Volume signature
<-	ioVDrvInfo	short	2	66	Drive number
<-	ioVDRefNum	short	2	68	Driver reference number
<-	ioVFSID	short	2	70	ID of file system handling this volume
<-	ioVBkUp	long	4	72	Date/time of most-recent backup
<-	ioVSeqNum	short	2	76	(used internally)
<-	ioVWrCnt	long	4	78	Volume write count
<-	ioVFilCnt	long	4	82	Count of files on entire volume
<-	ioVDirCnt	long	4	86	Count of directories on volume
<-	ioVFndrInfo[8]	long	32	90	Information used by the Finder

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
nsvErr	(-35)	No such volume
paramErr	(-50)	No default volume

Notes: **PBHGetVInfo** works just like **PBGetVInfo** except that it provides more return information. See that topic for a full discourse.

Differences:

- **PBHGetVInfo** always returns the volume reference number in ioVRefNum (**PBGetVInfo** might return a working directory number).
- The ioVNmAlBlks and ioVFrBlk fields are accurate for any size disk **(PBGetVInfo** clips these to an arbitrary maximum).

See **PBSetVInfo** for an example of usage.

The **PBHGetVInfo** call contains a little-known feature that allows you to get the volume names and vRefNum for all mounted volumes.

Example

```
OSErr GetIndVolume (short whichVol, char *volName, short *volRefNum)
{
    /* Return the name and vRefNum of volume specified by whichVol */

    HVolumeParam    volPB;
    OSErr           error;

    volPB.ioNamePtr = volName; /* make sure it returns the name */
    volPB.ioVRefNum = 0;      /* 0 means use ioVolIndex */
    volPB.ioVolIndex = whichVol;/* use this to determine volume */

    error = PBHGetVInfo(&volPB,false);  /* do it */
    if(error == noErr)
        *volRefNum = volPB.ioVRefNum;     /* return the volume reference */

    /* other information is available from this record; see the File Manager*/
    /* description of PBHGetVInfo for more details... */

    return (error);

}
```

This routine can be called several times to get information about all mounted volumes, starting with whichVol = 1, and incrementing whichVol until the routine returns nsvErr (-35, no such volume).