

GetVInfo

Get volume name, reference number and free bytes

#include <Files.h>

File Manager

<u>OSErr</u>	GetVInfo (<i>drvNum</i> , <i>volName</i> , <i>vRefNum</i> , <i>freeBytes</i>);	
<u>short</u>	<i>drvNum</i> ;	drive to query
<u>StringPtr</u>	<i>volName</i> ;	address of buffer to receive pascal-style name
<u>short</u>	* <i>vRefNum</i> ;	receives volume reference number
<u>long</u>	* <i>freeBytes</i> ;	receives amount of free space on the disk
	returns	<u>Error Code</u> ; 0=no error

Given a physical drive number, **GetVInfo** returns information about the volume mounted in that drive. Remember that since **GetVInfo** is only glue that fills in a parameter block for you and then calls **PBGetVInfo**, the values returned from it are subject to the limitations (imposed by MFS) of unsigned shorts for the ioVNmAlBks and ioVFrBk fields of the parameter block. If the actual numbers are larger than what fits in an unsigned short, they will be clipped to 31744.

drvNum identifies the physical drive of interest. Historically, 1=internal, 2=external, 3 and up are hard disks. However, in the SE and MacII, the drive numbers are assigned by the disk driver and may not fit this mold (*drvNum* =1 is always the first floppy drive).

volName is the address of a buffer. Upon return, the buffer will contain the volume name, as a length-prefixed, pascal-style string. The buffer should be at least 28 bytes long (to receive the 27-byte maximum volume name).

vRefNum is the address of an unsigned short. Upon return it will contain the volume's reference number.

freeBytes is the address of a long integer. Upon return it will contain the total free space (in bytes) available on the volume. This will be a multiple of the allocation block size for the volume.

Returns: an operating system Error Code. It will be one of:

<u>noErr</u>	(0)	No error
<u>nsvErr</u>	(-35)	No such volume
<u>paramErr</u>	(-50)	Bad <i>drvNum</i>

Notes: See **GetDrvQHdr** for a way to determine the drive numbers of all drives on the system. Use indexing techniques with **PBGetVInfo** to learn about all mounted volumes.

Example

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

```
/* for PtoCstr() */
```

```
Str255    volName;
short     vRef, rc;
long      avail;
```

```
rc = GetVInfo( 1, volName, &vRef, &avail );  
if ( rc ) { /* . . . process the error . . . */ }  
printf( "Drive 1: Volume Ref: %d; Bytes free: %ld, Name: '%s'\n",  
        vRef, avail, PtoCstr( volName ) );
```