UnmountVol Flush volume, close its files, release its memory

```
OSErr       UnmountVol(volName, vRefNum );
StringPtr   volName ;     address of Pascal-style name; NIL=use vRefNum
short       vRefNum ;     volume reference number
            returns       Error Code; 0=no error
```

UnmountVol flushes a volume buffer to disk and releases all memory occupied by the volume buffer and related structures. The volume must be re-mounted before it can be accessed.

volName is the address of a length-prefixed, pascal-style string containing the name of the volume you wish to flush. If *volName* is NIL (0), the *vRefNum* parameter is used.

vRefNum is the reference number of the volume you wish to flush. This parameter is used only if *volName* is invalid or NIL.

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

noErr	(0)	No error
bdNamErr	(-37)	Invalid *volName*
fBsyErr	(-47)	Files are open on volume
extFSErr	(-57)	External file system
ioErr	(-36)	I/O error
nsDrvErr	(-56)	No such drive
nsvErr	(-35)	No such volume
paramErr	(-50)	No default volume

Notes: Don't unmount the startup volume.

If you are using a string in the *volName* field to specify a volume name, the string must be in the following form

```
myVolParam.ioNamePtr = "\pMy HardDisk:"
```

The trailing colon on the string indicates that we are referring to a directory, not a file.

All files on a volume must be closed in order for **UnmountVol** to succeed. A fBsyErr will be received if this is not the case.

UnmountVol is typically called just before calling **Eject** - when the disk will not be needed again. Use **PBMountVol** to re-mount the volume, or (most commonly) let Standard File take care of mounting and unmounting.

A related call is **PBOffLine**, which retains the volume control block in memory so that **GetVInfo** will continue to return information and the volume can be brought back online transparently, by the File Manager, when needed.

THINK Reference © 1991-1992 Symantec Corporation