SetVol Select a new default volume or working directory

#include <Files.h> **File Manager**

```
OSErr       SetVol(volName, vRefNum );
StringPtr   volName ;     address of Pascal-style volume name string
short       vRefNum ;     volume or working directory reference
            returns       Error Code; 0=no error
```

SetVol selects a volume or working directory to become the default.

volName is the address of a length-prefixed, pascal-style string containing the name of the volume you wish to set as the default. Character case is ignored.

If *volName* is NIL (0), the *vRefNum* parameter will be used.

vRefNum is the reference number of the volume or working directory you wish to select as the new default. 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*
nsvErr	(-35)	No such volume
paramErr	(-50)	No default volume

Notes: **SetVol** lets you select a default volume for use in subsequent file operations where you do not specify a volume name or reference number. There is seldom any need for this since the Standard File Package functions return a volume reference indicating where a file is (or where the user wants it to go).

You can specify the desired volume by either a single name or a volume reference number; e.g.:

```
err = SetVol( 0, theRefNum );          /* set by reference number */

err = SetVol( "\pMy HardDisk:", 0 );          /* set by name */
```

The *volName* string should NOT be a multiple-name pathname (such as "\pHardDisk:Ltrs:Old") nor should it be an empty string (i.e., "\p"); this parameter is checked first and anything but a valid name (except a NIL pointer) is rejected as an error.

PBHSetVol lets you select both the default volume and default directory (see the second example, below).

You can pass a working directory number (i.e., the value of ioVRefNum after a call to **PBOpenWD**, or a volume reference returned by Standard File), but only if *volName* is NIL on entry (see the third example). Also, if you do use a working directory reference, a subsequent call to **GetVol** will return that number, rather than a "hard" volume ID.

Note: The "default volume" as used in **GetVol** and **SetVol** is NOT the same as the current volume or working directory used by the Standard File Package. Those values are maintained in the global variables SFSaveDisk and CurDirStore and they may have no relationship with the volume (or directory) you wish to single out as your application default. See **SFGetFile**, et al.

The following example uses several variations of **SetVol** and **FSOpen** to illustrate different ways of referring to a file:

Example

```
#include <Files.h>

WDPBRec  wdpb;          /* for PBHSetVol and PBOpenWD */
short   fRefNum;

/* First example: ------------------------------------
    Set the default volume and use partial pathname to open the file.
*/
SetVol( "\pHardDisk:", 0 );
FSOpen( "\p:Ltrs:1988:Jones", 0, &fRefNum );

/* Second example: -----------------------------------
    Set default volume and a partial directory via PBHSetVol and
    use a partial pathname to open the file.
*/
wdpb.ioNamePtr = (StringPtr)"\pHardDisk:Ltrs:";
PBHSetVol( &wdpb, FALSE );
FSOpen( "\p:1988:Jones", 0, &fRefNum );     /* partial name */

/* Third example: ------------------------------------
    Open a working directory, set it as the default, and use a
    partial pathname (including a "parent" directory) to open the file.
*/

wdpb.ioNamePtr = (StringPtr)"\pHardDisk:Ltrs:1987:";
wdpb.ioWDDirID = 0;             /* not using a "hard" directory ID */
PBOpenWD( &wdpb, FALSE );
SetVol( 0, wdpb.ioVRefNum );        /* must use NIL (0) for name */
FSOpen( "\p::1988:Jones", 0, &fRefNum );            /* :: is parent */
```