PBGetWDInfo Query information about a working directory

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

```
OSErr     PBGetWDInfo(pb, async);
WDPBPtr   pb ;        address of a 52-byte WDPBRec structure
Boolean   async ;     0=await completion; 1=immediate return
          returns     Error Code; 0=no error
```

PBGetWDInfo obtains the "hard" directory ID and the real volume number associated with a working directory. This function also lets you index through the list of open working directory control blocks.

pb is the address of a 52-byte WDPBRec structure (or any other structure) with the following fields:

Out-In	Name	Type	Size	Offset	Description
–>	ioCompletion	ProcPtr	4	12	Completion routine address (if *async* =TRUE)
–>	ioWDIndex	short	2	26	Index (indexed searches); 0=lookup ioVRefNum
<->	ioVRefNum	short	2	22	Entry: Working dir ref, or vol to index (0=all) Returns: Working dir ref, or WD's "hard" volume
<->	ioWDProcID	long	4	28	Entry: Working dir user refs to index (0=any) Return: User ref of WD (if indexing)
<->	ioWDVRefNum	short	2	32	Volume in which working directory is located
<-	ioResult	OSErr	2	16	Error Code (0=no error, 1=not done yet)
<-	ioNamePtr	StringPtr	4	18	Addr of buffer to hold 28-byte max volume name
<-	ioWDDirID	long	4	48	Working directory's "hard" directory ID

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

Notes: You can use **PBGetWDInfo** after calling **SFGetFile** to get the ioDirID to use in subsequent **PBH**xxx calls (you could also just look in the global variable CurDirStore).

To index through the WD control block chain, set ioWDIndex to successively higher numbers until the function returns an error. If you wish to examine all WD blocks, be sure to set ioWDProcID to 0 before each call. Also, you can set ioVRefNum to a "hard" volume number to limit the indexing to only those working directories associated with a particular disk.

The ioNamePtr field should point to a 28-byte minimum buffer. Upon return, it will contain a pascal-style string of the volume name (root directory name) associated with a working directory. There is no direct way to obtain a directory's fully-qualified (multiple-name) pathname.

Example

```
#include <Files.h>
#include <StandardFile.h>        // for SFReply structure
```

```
SFReply    reply;
WDPBRec  wdpb;
Point    where;
char     volName[28];          // buffer to hold name
long     myDirID;

where.h=100; where.v=50;
SFGetFile( where, "\pthe prompt", 0, -1, 0, 0, &reply );

if ( reply.good ) {
    wdpb.ioNamePtr = volName;  // use NIL (0) if don't care
    wdpb.ioVRefNum = reply.vRefNum;
    wdpb.ioWDIndex =0;             // not indexing here
    wdpb.ioWDProcID =0;                // don't care
    wdpb.ioWDVRefNum =0;       // don't care
    PBGetWDInfo( &wdpb, FALSE );
    myDirID = wdpb.ioWDDirID; // save for use later
}
```