GetCurrentDirectory function

Retrieves the current directory for the current process.

Syntax

C++

```
DWORD WINAPI GetCurrentDirectory(
  _In_  DWORD  nBufferLength,
  _Out_ LPTSTR lpBuffer
);
```

Parameters

nBufferLength [in]

The length of the buffer for the current directory string, in **TCHARs**. The buffer length must include room for a terminating null character.

lpBuffer [out]

A pointer to the buffer that receives the current directory string. This null-terminated string specifies the absolute path to the current directory.

To determine the required buffer size, set this parameter to **NULL** and the *nBufferLength* parameter to 0.

Return value

If the function succeeds, the return value specifies the number of characters that are written to the buffer, not including the terminating null character.

If the function fails, the return value is zero. To get extended error information, call **GetLastError**.

If the buffer that is pointed to by *lpBuffer* is not large enough, the return value specifies the required size of the buffer, in characters, including the null-terminating character.

Remarks

Each process has a single current directory that consists of two parts:

- A disk designator that is either a drive letter followed by a colon, or a server name followed by a share name

(*servername**sharename*)
- A directory on the disk designator

To set the current directory, use the SetCurrentDirectory function.

Multithreaded applications and shared library code should not use the **GetCurrentDirectory** function and should avoid using relative path names. The current directory state written by the SetCurrentDirectory function is stored as a global variable in each process, therefore multithreaded applications cannot reliably use this value without possible data corruption from other threads that may also be reading or setting this value. This limitation also applies to the **SetCurrentDirectory** and GetFullPathName functions. The exception being when the application is guaranteed to be running in a single thread, for example parsing file names from the command line argument string in the main thread prior to creating any additional threads. Using relative path names in multithreaded applications or shared library code can yield unpredictable results and is not supported.

In Windows 8 and Windows Server 2012, this function is supported by the following technologies.

Technology	Supported
Server Message Block (SMB) 3.0 protocol	Yes
SMB 3.0 Transparent Failover (TFO)	Yes
SMB 3.0 with Scale-out File Shares (SO)	Yes
Cluster Shared Volume File System (CsvFS)	Yes
Resilient File System (ReFS)	Yes

Examples

For an example, see Changing the Current Directory.

Requirements

Minimum supported client	Windows XP [desktop apps | Windows Store apps]
Minimum supported server	Windows Server 2003 [desktop apps | Windows Store apps]
Header	WinBase.h (include Windows.h)

Library	Kernel32.lib
DLL	Kernel32.dll
Unicode and ANSI names	**GetCurrentDirectoryW** (Unicode) and **GetCurrentDirectoryA** (ANSI)

See also

CreateDirectory
Directory Management Functions
GetSystemDirectory
GetWindowsDirectory
RemoveDirectory
SetCurrentDirectory