WinExec function (winbase.h) - Win32 apps

By GrantMeStrength

Archived: 2026-04-05 22:57:54 UTC

Runs the specified application.

Note This function is provided only for compatibility with 16-bit Windows. Applications should use the CreateProcess function.

```
UINT WinExec(
  [in] LPCSTR lpCmdLine,
  [in] UINT   uCmdShow
);
```

`[in] lpCmdLine`

The command line (file name plus optional parameters) for the application to be executed. If the name of the executable file in the *lpCmdLine* parameter does not contain a directory path, the system searches for the executable file in this sequence:

1. The directory from which the application loaded.
2. The current directory.
3. The Windows system directory. The GetSystemDirectory function retrieves the path of this directory.
4. The Windows directory. The GetWindowsDirectory function retrieves the path of this directory.
5. The directories listed in the PATH environment variable.

`[in] uCmdShow`

The display options. For a list of the acceptable values, see the description of the *nCmdShow* parameter of the ShowWindow function.

If the function succeeds, the return value is greater than 31.

If the function fails, the return value is one of the following error values.

Return code/value	Description
0	The system is out of memory or resources.
ERROR_BAD_FORMAT	The .exe file is invalid.

ERROR_FILE_NOT_FOUND	The specified file was not found.
ERROR_PATH_NOT_FOUND	The specified path was not found.

The **WinExec** function returns when the started process calls the GetMessage function or a time-out limit is reached. To avoid waiting for the time out delay, call the **GetMessage** function as soon as possible in any process started by a call to **WinExec**.

The executable name is treated as the first white space-delimited string in *lpCmdLine*. If the executable or path name has a space in it, there is a risk that a different executable could be run because of the way the function parses spaces. The following example is dangerous because the function will attempt to run "Program.exe", if it exists, instead of "MyApp.exe".

```
WinExec("C:\\Program Files\\MyApp", ...)
```

If a malicious user were to create an application called "Program.exe" on a system, any program that incorrectly calls **WinExec** using the Program Files directory will run this application instead of the intended application.

To avoid this problem, use CreateProcess rather than **WinExec**. However, if you must use **WinExec** for legacy reasons, make sure the application name is enclosed in quotation marks as shown in the example below.

```
WinExec("\"C:\\Program Files\\MyApp.exe\" -L -S", ...)
```

Requirement	Value
Minimum supported client	Windows XP [desktop apps only]
Minimum supported server	Windows Server 2003 [desktop apps only]
Target Platform	Windows
Header	winbase.h (include Windows.h)
Library	Kernel32.lib
DLL	Kernel32.dll
API set	ext-ms-win-kernel32-process-l1-1-0 (introduced in Windows 10, version 10.0.14393)

CreateProcess
