# ShellExecute

Quick Info | Overview | Group

The **ShellExecute** function opens or prints a specified file. The file can be an executable file or a document file. See **ShellExecuteEx** also.

```
HINSTANCE ShellExecute(
  HWND hwnd,              // handle to parent window
  LPCTSTR lpOperation,    // pointer to string that specifies operation to perform
  LPCTSTR lpFile,         // pointer to filename or folder name string
  LPCTSTR lpParameters,   // pointer to string that specifies executable-file parameters
  LPCTSTR lpDirectory,    // pointer to string that specifies default directory
  INT nShowCmd            // whether file is shown when opened
);
```

## Parameters

*hwnd*

Specifies a parent window. This window receives any message boxes that an application produces. For example, an application may report an error by producing a message box.

*lpOperation*

Pointer to a null-terminated string that specifies the operation to perform. The following operation strings are valid:

| String | Meaning |
|---|---|
| "open" | The function opens the file specified by *lpFile*. The file can be an executable file or a document file. The file can be a folder to open. |
| "print" | The function prints the file specified by *lpFile*. The file should be a document file. If the file is an executable file, the function opens the file, as if "open" had been specified. |
| "explore" | The function explores the folder specified by *lpFile*. |

The *lpOperation* parameter can be NULL. In that case, the function opens the file specified by *lpFile*.

*lpFile*

Pointer to a null-terminated string that specifies the file to open or print or the folder to open or explore. The function can open an executable file or a document file. The function can print a document file.

*lpParameters*

If *lpFile* specifies an executable file, *lpParameters* is a pointer to a null-terminated string that specifies parameters to be passed to the application.

If *lpFile* specifies a document file, *lpParameters* should be NULL.

*lpDirectory*

Pointer to a null-terminated string that specifies the default directory.

*nShowCmd*

If *lpFile* specifies an executable file, *nShowCmd* specifies how the application is to be shown when it is opened. This parameter can be one of the following values:

| Value | Meaning |
|---|---|
| SW_HIDE | Hides the window and activates another window. |
| SW_MAXIMIZE | Maximizes the specified window. |
| SW_MINIMIZE | Minimizes the specified window and activates the next top-level window in the Z order. |
| SW_RESTORE | Activates and displays the window. If the window is minimized or maximized, Windows restores it to its original size and position. An application should specify this flag when restoring a minimized window. |

| | |
|---|---|
| SW_SHOW | Activates the window and displays it in its current size and position. |
| SW_SHOWDEFAULT | Sets the show state based on the SW_ flag specified in the **STARTUPINFO** structure passed to the **CreateProcess** function by the program that started the application. An application should call **ShowWindow** with this flag to set the initial show state of its main window. |
| SW_SHOWMAXIMIZED | Activates the window and displays it as a maximized window. |
| SW_SHOWMINIMIZED | Activates the window and displays it as a minimized window. |
| SW_SHOWMINNOACTIVE | Displays the window as a minimized window. The active window remains active. |
| SW_SHOWNA | Displays the window in its current state. The active window remains active. |
| SW_SHOWNOACTIVATE | Displays a window in its most recent size and position. The active window remains active. |
| SW_SHOWNORMAL | Activates and displays a window. If the window is minimized or maximized, Windows restores it to its original size and position. An application should specify this flag when displaying the window for the first time. |

If *lpFile* specifies a document file, *nShowCmd* should be zero.

## Return Values

If the function succeeds, the return value is the instance handle of the application that was run, or the handle of a dynamic data exchange (DDE) server application.

If the function fails, the return value is an error value that is less than or equal to 32. The following table lists these error values:

| Value | Meaning |
|---|---|
| 0 | The operating system is out of memory or resources. |
| ERROR_FILE_NOT_FOUND | The specified file was not found. |
| ERROR_PATH_NOT_FOUND | The specified path was not found. |
| ERROR_BAD_FORMAT | The .EXE file is invalid (non-Win32 .EXE or error in .EXE image). |
| SE_ERR_ACCESSDENIED | The operating system denied access to the specified file. |
| SE_ERR_ASSOCINCOMPLETE | The filename association is incomplete or invalid. |
| SE_ERR_DDEBUSY | The DDE transaction could not be completed because other DDE transactions were being processed. |
| SE_ERR_DDEFAIL | The DDE transaction failed. |
| SE_ERR_DDETIMEOUT | The DDE transaction could not be completed because the request timed out. |
| SE_ERR_DLLNOTFOUND | The specified dynamic-link library was not found. |
| SE_ERR_FNF | The specified file was not found. |
| SE_ERR_NOASSOC | There is no application associated with the given filename extension. |
| SE_ERR_OOM | There was not enough memory to complete the |

| | |
|---|---|
| | operation. |
| SE_ERR_PNF | The specified path was not found. |
| SE_ERR_SHARE | A sharing violation occurred. |

## Remarks

The file specified by the *lpFile* parameter can be a document file or an executable file. If the file is a document file, the **ShellExecute** function opens or prints it, depending on the value of the *lpOperation* parameter. If the file is an executable file, the **ShellExecute** function opens it, even if *lpOperation* specifies printing.

You can use **ShellExecute** to open or explore a shell folder. To open a folder, use either of the following calls:

```
ShellExecute(handle, NULL, path_to_folder, NULL, NULL, SW_SHOWNORMAL);
  or

ShellExecute(handle, "open", path_to_folder, NULL, NULL, SW_SHOWNORMAL);
```

To explore a folder, use the following call:

```
ShellExecute(handle, "explore", path_to_folder, NULL, NULL, SW_SHOWNORMAL);
```

If *lpOperation* is NULL, the function opens the file specified by *lpFile*. If *lpOperation* is "open" or "explore", the function will force an open window or explorer.

## See Also

**FindExecutable**, **ShellExecuteEx**