# GetLastError function

Retrieves the calling thread's last-error code value. The last-error code is maintained on a per-thread basis. Multiple threads do not overwrite each other's last-error code.

**Visual Basic:** Applications should call **err.LastDllError** instead of **GetLastError**.

## Syntax

C++

```
DWORD WINAPI GetLastError(void);
```

## Parameters

This function has no parameters.

## Return value

The return value is the calling thread's last-error code.

The Return Value section of the documentation for each function that sets the last-error code notes the conditions under which the function sets the last-error code. Most functions that set the thread's last-error code set it when they fail. However, some functions also set the last-error code when they succeed. If the function is not documented to set the last-error code, the value returned by this function is simply the most recent last-error code to have been set; some functions set the last-error code to 0 on success and others do not.

## Remarks

Functions executed by the calling thread set this value by calling the SetLastError function. You should call the **GetLastError** function immediately when a function's return value indicates that such a call will return useful data. That is because some functions call **SetLastError** with a zero when they succeed, wiping out the error code set by the most recently failed function.

To obtain an error string for system error codes, use the FormatMessage function. For a complete list of error codes provided by the operating system, see System Error Codes.

The error codes returned by a function are not part of the Windows API specification and can vary by operating system or device driver. For this reason, we cannot provide the complete list of error codes that can be returned by each function. There are also many functions whose documentation does not include even a partial list of error codes that can be returned.

Error codes are 32-bit values (bit 31 is the most significant bit). Bit 29 is reserved for application-defined error codes; no system error code has this bit set. If you are defining an error code for your application, set this bit to one. That indicates that the error code has been defined by an application, and ensures that your error code does not conflict with any error codes defined by the system.

To convert a system error into an **HRESULT** value, use the HRESULT_FROM_WIN32 macro.

## Examples

For an example, see Retrieving the Last-Error Code.

## Requirements

| | |
|---|---|
| **Minimum supported client** | Windows XP [desktop apps \| Windows Store apps] |
| **Minimum supported server** | Windows Server 2003 [desktop apps \| Windows Store apps] |
| **Minimum supported phone** | Windows Phone 8 |
| **Header** | WinBase.h (include Windows.h) |
| **Library** | Kernel32.lib |
| **DLL** | Kernel32.dll |

## See also

Error Handling Functions
FormatMessage
HRESULT_FROM_WIN32
Last-Error Code
SetLastError
SetLastErrorEx

# Community Additions

## An GetLastErrorEx() would be nice

An GetLastErrorEx() would be nice, which gives more detail about what went wrong.
For example for "RegisterClassEx()" it could return something like:
"Parameter lpwcx is invalid. WNDCLASSEX-structure has invalid value for cbSize"

thegeeky
7/23/2015

## Some useful C code

```
// usage
//     CHAR msgText[256];
//     getLastErrorText(msgText,sizeof(msgText));
static CHAR *                   //   return error message
getLastErrorText(               // converts "Lasr Error" code into text
CHAR *pBuf,                     //   message buffer
ULONG bufSize)                    //   buffer size
{
    DWORD retSize;
    LPTSTR pTemp=NULL;

    if (bufSize < 16) {
        if (bufSize > 0) {
            pBuf[0]='\0';
        }
        return(pBuf);
    }
    retSize=FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER|
                          FORMAT_MESSAGE_FROM_SYSTEM|
                          FORMAT_MESSAGE_ARGUMENT_ARRAY,
                          NULL,
                          GetLastError(),
                          LANG_NEUTRAL,
                          (LPTSTR)&pTemp,
                          0,
                          NULL );
    if (!retSize || pTemp == NULL) {
        pBuf[0]='\0';
    }
    else {
        pTemp[strlen(pTemp)-2]='\0'; //remove cr and newline character
        sprintf(pBuf,"%0.*s (0x%x)",bufSize-16,pTemp,GetLastError());
        LocalFree((HLOCAL)pTemp);
    }
    return(pBuf);
}
```


Infinetwork
9/29/2014

---

### fasf

fdsfsf


abdalla 1235
10/6/2013

---

### CLastError

Here is a C++ class that passed ::GetLastError() code will retrieve or display user friendly text associated with the error code: http://www.paskov.biz/blog/clasterror/


Vladimir Paskov
8/31/2012

---

### err.LastDllError

DWORD WINAPI GetLastError(void);


maffiemiauw
8/3/2012

---

### sy6stem 32 restore

C:\Windows\System32\restore


Jeffrey Ahrens
5/14/2012

---

### getlasterror

DWORD WINAPI GetLastError(void);


izzy sablan
4/30/2012

---

### Free Pascal 2.4.4

Free Pascal, Windows API, GetLastError

Free Pascal 2.4.4

extracted from Windows unit "Windows.ppu"

```
function GetLastError
  : LongWord;
  stdcall;
```

armydudematt
3/23/2012

### May incorrectly return 0 within VS2008 debugger

In VS2008 GetLastError may incorrectly always return zero if you are performing mixed mode debugging (Native + Managed) and s ingle stepping.

The workaround is to:

- Do native only debugging in areas involving GetLastError() calls OR
- Cleverly set breakpoints so you don't single step through GetLastError() calls OR
- Use VS2010

Psychopharm7
6/2/2011

### Don't call through PInvoke

http://blogs.msdn.com/adam_nathan/archive/2003/04/25/56643.aspx

Adam Nathan indicates you should use Marshal.GetLastWin32Error() instead.

Buddha64
3/24/2010

### Possible VB9 declaration

```
Public Declare Function GetLastError Lib "kernel32" () As Integer
In fact there is no need to use this function from both - VB6 and .NET.

From .NET you can simply create instance of Win32Exception immediately after API call and you get all the
information about error that had occurred (if any).
```

Đonny
12/7/2007