

How can I tell whether my console program was launched from Explorer or from a command prompt?

 devblogs.microsoft.com/oldnewthing/20160125-00

January 25, 2016



Raymond Chen

So you have a console program. This console program could be run by typing its name at the command prompt, or it could be run by the user double-clicking it from Explorer. And you want to know which case you're in.

This is another case of digging into the question to find the problem. In this case, the problem is "Well, if I'm run directly from Explorer, then when my program exits, the console is destroyed with it, and the user can't see the output. In that case, I want to prompt the user to hit Enter before the program exits."

Okay, so what you really want to know is not whether you were run from Explorer. (After all, you would have this problem if the program were run from Task Manager or some other program launcher.) What you really want to know is whether the console will continue to exist after your program exits.

For that, you can use `GetConsoleProcessList` .

If your process is the only one attached to the console, then the console will be destroyed when your process exits. If there are other processes attached to the console, then the console will continue to exist (because your program won't be the last one).

```

#include <windows.h>
#include <stdio.h>

int __cdecl wmain()
{
    printf("this process = %d\n", GetCurrentProcessId());
    DWORD count = GetConsoleProcessList(nullptr, 0);
    if (count == 1) {
        printf("I'm the last one!\n");
        Sleep(2000);
    } else {
        printf("I'm not the last one!\n");
    }
    return 0;
}

```

We care only how many processes are in the console process list; we don't care what they are. After getting the count, we either declare that we're the last one (and pause so you can see the message), or we say that we aren't (and exit immediately).

This is accurate as far as it goes: It tells you whether the console will be destroyed when the process exits. What it doesn't tell you is whether the other processes in the process list will also exit when you exit. For example, if somebody does

```
start cmd /c scratch.exe
```

then the program will correctly report that it's not the last one, but what it doesn't know is that the `cmd.exe` is going to exit as soon as the `scratch.exe` program exits. There's not much you can do to detect this, because that information is internal to whatever other process launched your program.

[Raymond Chen](#)

Follow

