

# Why does my program successfully take foreground only when running under the debugger?

 devblogs.microsoft.com/oldnewthing/20241128-00

November 28, 2024



A customer found that their program was inconsistent in whether it was able to take foreground. They figured out that the program succeeded only if they ran it from the Visual Studio debugger. If they ran the program outside the debugger, the window merely blinked in the taskbar.

Why does running in the debugger change the behavior?

If you look up [the rules for SetForegroundWindow](#), you'll see this as one of the options for allowing a process to take foreground:

| Either the foreground process or the calling process is being debugged.

This rule exists because the debugger itself takes foreground when a breakpoint is reached, and the system doesn't want to punish the program being debugged.

Unfortunately, it means that this extra leniency can make you think that your program is properly managing foreground when in fact it is getting a special pass because it is being debugged.

Without a debugger, you need to stick with the normal foreground rules.

**Related reading:** Foreground activation permission is like love: You can't steal it, it has to be given to you.