

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

devblogs.microsoft.com/oldnewthing/20090220-00

February 20, 2009



Raymond Chen

This is the blog entry that acted as the inspiration for the last topic in [my 200 PDC talk](#).

When somebody launches a second copy of your single-instance program, you usually want the second copy to send its command line to the first instance (and deal with the current directory somehow), and then you want the first instance to come to the foreground. But a common problem people run into is that when the first instance calls `SetForegroundWindow`, it fails.

The problem with this design is that as far as the window manager is concerned, what happened is that the first instance received a message and then decided to steal foreground. That message wasn't an input message, so the window manager sees no reason for the first instance to have any right to take the foreground. There is no evidence that the first instance is coming to the foreground in response to some user action.

There are a variety of ways of addressing this problem. The easiest way is simply to have the second instance make the call to `SetForegroundWindow`. The second program has permission to take the foreground because you just launched it. And if a program can take the foreground, it can also give it away, in this case, by setting the first program as the foreground window.

Another way to do this is to have the second program call the `AllowSetForegroundWindow` function with the process ID of the first program before it sends the magic message. The `AllowSetForegroundWindow` function lets a program say to the window manager, "It's okay; he's with me." And then when the first program finally gets around to calling `SetForegroundWindow`, the window manager says, "Oh, this guy's okay. That other program vouched for him."

If you are transferring foreground activation to a COM server, you can use the corresponding COM function `CoAllowSetForegroundWindow`.

In all cases, note that you can't give away something that's not yours. If you don't have permission to take foreground, then calling `AllowSetForegroundWindow` (or one of its moral equivalents) will have no effect: You just told the window manager, "It's okay; he's with me," and the window manager replied, "Who the hell are you?"

Pre-emptive snarky comment: "There are some really sneaky people who found a way to circumvent the rules and steal foreground activation." Well yeah, and there are some really sneaky people who find ways to steal love, so there you have it. If everything is right with the world, both groups of people will eventually be found out and made to suffer for their malfeasance.

Update (02/21): Deleted all comments that showed ways of circumventing the rules. Duh, people.

Raymond Chen

Follow

