

# You don't need to steal focus if you can just arrange for someone to give it to you

[devblogs.microsoft.com/oldnewthing/20110613-00](http://devblogs.microsoft.com/oldnewthing/20110613-00)

June 13, 2011



Raymond Chen

Commenter *doesn't matter* proposes a scenario where focus-stealing is necessary: There are two applications A and B, Application B exposes an interface, and Application A connects to application B via that interface, When some sort of even occurs in application B, it notifies application A, which wants to steal focus in order to interact with the user as a result of the event. Actually, this is still not a situation where focus-stealing is necessary. Application B just needs to call `AllowSetForegroundWindow` on application A before it fires the event, so that when application A receives the event, it can display its dialog box or whatever. If the communication channel is via a window message, you can use `GetWindowThreadProcessId` to get the process ID of application A. If the communication channel is COM, you can use `CoAllowSetForegroundWindow`. If the communication channel is something else, well, then you'll have to use whatever mechanism works for your communication channel. (Obviously there must be at some point a place in the communication channel where it can figure out the identity of application A, since it needs to talk to application A to deliver the event in the first place.) If you don't have access to the source code of application B, then you get to work with whoever designed the interface to convince them to add the necessary call for you. But outright stealing the focus is the wrong thing to do, because it presumes that the user was interacting with application B when the event was fired. What if the event fired while application B was not the focus? Even if the event is a user interface event, like a mouse click, it's possible that the event will fire even though application B doesn't have the focus: Application B may receive the mouse click, and while application B decides what to do with it, the user switches to application C. Eventually, application B fires the event, but at this point the user is no longer working with application B, and transferring focus to application A would count as one of those evil focus-stealing situations. Just because there's no good way to do something doesn't mean that you are automatically permitted to do it in a bad way. **Bonus chatter:** Sometimes I wonder about the people who use the principle *If I can't do something legally, I should be allowed to do it illegally*, and then get indignant when they are caught. "You got a parking ticket because you were parked in a handicapped space without a permit." — *But all the regular parking spaces were taken. I had to park in a handicapped space; I had no other choice. You can't give me a ticket for that!*

[Raymond is currently away; this message was pre-recorded.]

Raymond Chen

**Follow**

