Be careful that your splash screen doesn't squander the foreground love

devblogs.microsoft.com/oldnewthing/20100823-00

August 23, 2010

Raymond Chen

Commenter Erbi has a program which creates a splash screen on a background thread while the main thread initializes. "I create and then destroy this splash screen window just before creating and displaying the main window." The problem is that the main window fails to obtain foreground activation. Commenting out the code that creates the splash screen fixes the problem, but then there isn't a splash screen any more (obviously). "Is there an explanation for this behavior?" This behavior is explained by two earlier blog posts, plus a PDC talk. The first blog post came out years before this question was asked: The correct order for disabling and enabling windows. Destroying a window is a rather extreme case of disabling it, but the effect is the same. When you destroy the splash screen, foreground activation needs to move to some other window, and since your main window isn't around to inherit it, foreground activation leaves your program. When the main window appears, it's too late.

The PDC talk came next, followed shortly thereafter by a blog post version of the same talk. As marketing folks like to remind you, "You get only one chance to make a first impression." Similarly, you get only one chance to use your foreground activation permission, and you decided to blow it on a splash screen. That's fine as far as it goes, but if you want to transfer that permission to another window, you have to manage it yourself. The recommended way is to establish an owner/owned relationship between them; that's the case that the "disabling and enabling windows" article focuses on.