

I used `WS_EX_COMPOSITED` to get rid of my redrawing flicker, but it resulted in sluggish response

 devblogs.microsoft.com/oldnewthing/20171018-00

October 18, 2017



Raymond Chen

A customer enabled `the WS_EX_COMPOSITED extended window style` in order to solve a flickering problem with their application. That worked great, but they found that it had an unwanted side effect: when the user grabs the scroll bar and starts scrolling up and down the document, the UI does not update smoothly. The customer hoped for some tips on how they could get the no-flicker benefit of `WS_EX_COMPOSITED` without incurring the sluggish-UI penalty.

One of my colleagues explained: The `WS_EX_COMPOSITED` extended style enables double-buffering on the window. This means that the control renders into an off-screen buffer, and only when rendering is complete is the result copied to the screen. This avoids flicker because only a completely-drawn control is put on the screen; you never see the control in a partially-drawn state.

The thing is, in order for this to work, the system needs to know when rendering is complete. Otherwise it doesn't know when it's time to copy the back-buffer to the screen.

The rule is that the system assumes that you're done rendering when you release the DC. (Or when you call `EndPaint`, which is the moral equivalent of releasing the DC.)

But there are some programs which acquire a DC and never release it. They just draw into it whenever they feel like drawing. In that case, the system has a fail-safe timeout and copies the back-buffer to the screen after 100ms, and again every 100ms thereafter.

If you're getting slugging UI updates with the `WS_EX_COMPOSITED` extended window style, check that you're properly releasing your device contexts as soon as you're ready for them to be presented. Don't acquire a DC and just hang onto it for convenience. If you do that, then the system doesn't know when you're finished drawing, and it will copy from the back-buffer to the screen at 10 frames per second, which will feel sluggish.

[Raymond Chen](#)

Follow

