

# Final remarks on LockWindowUpdate

 [devblogs.microsoft.com/oldnewthing/20070223-00](http://devblogs.microsoft.com/oldnewthing/20070223-00)

February 23, 2007



Raymond Chen

Now that you understand the intended purpose of `LockWindowUpdate`, I'm going to tell you why you don't want to use it, not even for its intended purpose!

You have to go back to the historical context in which `LockWindowUpdate` was created. Rewind back to 16-bit Windows, specifically Windows 3.1. Back in these days, memory was expensive. Video drivers were pretty limited in functionality. There was no DirectX. There was no AlphaBlend function. All you had was a screen buffer. The `LockWindowUpdate` function let you take control over one window's portion of that screen buffer so you could apply your fancy effects to the window without that window's knowledge.

It's been over a decade since Windows 3.1, and in the meanwhile, we gained DirectX overlays, regional windows, layered windows, alpha blending, desktop composition, all sorts of cool graphical effects that weren't available back in the old days. In particular, those layered windows and regional windows pretty much let you do nearly all of the stuff you would have wanted to do with `LockWindowUpdate` anyway. If you want to draw a highlight around a window, you can position a regional window around it. If you want to draw a drag image over a window, you can just create layered window and position it over the target window. Give the layered window a region and whatever fancy alpha channel you want, and let the graphics engine do the heavy lifting of alpha blending and composition. Even better, the layered window can extend outside the window you are dragging over, something that `LockWindowUpdate` can't do. (You can see this effect in Windows XP if you do a "select all" in an Explorer window and drag the entire selection around the screen. Notice that the drag image is not constrained to the boundaries of the window you are dragging over.)

What's more, in the exciting new composited world of Vista's desktop window manager, `LockWindowUpdate` is even less desirable. Locking a particular window for update isn't so bad since the desktop window manager can just give you the backing bitmap for the window. But if you lock the entire screen (which I've seen many people do), the desktop window manager needs to compose all of the windows into an *actual bitmap* that it can give you when you call `GetDCEX` with the `DCX_LOCKWINDOWUPDATE` flag. The desktop window manager does composition on the fly with the help of DirectX and accelerated video drivers. The result of all this composition normally goes straight to the screen without actually

residing in a “composited” bitmap anyway. But if you lock the screen and ask for a DC to it, the desktop window manager needs to emulate the old behavior and give you access to something that represents what you *would have gotten* if there were no composition in the first place. That ain’t cheap.

**Epilogue.** I’m not sure if this series was a success or not. My goal was just to help people use `LockWindowUpdate` more effectively and guide them towards alternatives when `LockWindowUpdate` is the wrong tool for the job. In other words, it’s an article about `LockWindowUpdate`, not function documentation. I tried to keep the presentation light, but I guess my jokes fell flat, and people just used them as a springboard for negative comments.

And extra thanks to the people who took it as an opportunity to complain about the documentation. I mean, duh, if the documentation were perfect, I wouldn’t have written this series in the first place. Though these people also neglected to read *all* of the documentation; they looked only at the function description page. There’s more to documentation than dry function descriptions, people! The function description is a *reference*; you go there when you already know what’s going on and you just need to fine-tune a detail. The real learning happens in the overviews and articles. If you want to learn how to operate your radio, you don’t read the schematic first.

I think Ronald D. Moore is really onto something when he says, “You have to be tough enough to listen to the podcast.”

[Raymond Chen](#)

**Follow**

