

With what operations is LockWindowUpdate meant to be used?

devblogs.microsoft.com/oldnewthing/20070221-11

February 21, 2007



Raymond Chen

As I noted earlier, the intended purpose of `LockWindowUpdate` can be captured in one word: dragging. The simplest case of `LockWindowUpdate` is used by the window manager when you move or resize a window and “Show window contents while dragging” is disabled. When you start the move/size operation, the window manager locks the entire desktop so it can draw the dotted-rectangle feedback without risking interference from another window that happens to intersect the dotted-rectangle. When the move/size operation is complete, the desktop is unlocked, and the world returns to normal. A common case where an application uses `LockWindowUpdate` is if it wants to draw a custom image when offering drag feedback. In this case, the application locks **its own window** in order to draw the drag feedback. It then uses the `DCX_LOCKWINDOWUPDATE` flag to get a DC that it can use to draw the desired feedback, and it doesn’t have to worry about the window procedure or any other code in the application accidentally drawing to the feedback window and messing up the drag image. For example, if the application is displaying drag feedback on a list view, and some asynchronous event caused the list view contents to change (say, an item got added), and the drag image just happens to be where the new item is about to appear, you wouldn’t want the normal redraw behavior of the listview to overpaint (or worse, merge with) the drag image. A case where you would want to lock another application’s window is if you are dragging an object across the screen. You might do this if you are a program like Spy that has an option to let the user select a window by dragging a “selection gadget” over it. Not only do you have to lock the window the user is selecting so that its own painting won’t conflict with your “selection gadget”, but also so that it doesn’t conflict with the highlight effect you place around that window.

By now, you’ve probably noticed a common thread to all of these `LockWindowUpdate` scenarios: They all involve dragging of some sort. Dragging a window caption to move it, dragging the edge of a window to resize it, dragging an item into a window, or dragging an item out of a window. This is not a coincidence. The `LockWindowUpdate` function was designed with these drag scenarios in mind. Since dragging an item uses the mouse button,

and there's only one mouse, you can't have multiple drag operations in progress at once, and therefore, there's no need to lock more than one window for update at a time. The function should perhaps more accurately have been named `LockDragWindow` .

Raymond Chen

Follow

