

# Speeding up adding items to a combobox or listbox

 [devblogs.microsoft.com/oldnewthing/20040610-00](http://devblogs.microsoft.com/oldnewthing/20040610-00)

June 10, 2004



Raymond Chen

Just a little tip: If you're going to be adding a lot of items to a listbox or combobox, there are a few little things you can do to improve the performance significantly. (Note: The improvements work only if you have a **lot** of items, like hundreds. Of course, the usability of a listbox with a hundred items is questionable, but I'm assuming you have a **really good reason** for doing this.)

First, you can disable redraw while you add the items. (This tip works for all controls, actually.)

```
SetWindowRedraw(hwnd, FALSE);  
... add the items ...  
SetWindowRedraw(hwnd, TRUE);  
InvalidateRect(hwnd, NULL, TRUE);
```

`SetWindowRedraw` is defined in the `<windowsx.h>` header file.

This suppresses the control redrawing itself each time you add an item. But there is still something else you can do:

```
SendMessage(hwndCombo, CB_INITSTORAGE, cItems, cbStrings);  
... add the items ...
```

(For listbox controls, use the `LB_INITSTORAGE` message.)

`cItems` is the number of items you intend to add, and `cbStrings` is the total amount of memory (in bytes) required by the strings you intend to add.

It's okay if these values are merely estimates. If you are too low, then there will still be some reallocation for the extra items. If you are too high, then some memory will be allocated but remain unused.

Some people will recommend using `LockWindowUpdate`, but this is **wrong**.

`LockWindowUpdate` disables drawing in the window you specify, but suppressing flickering during updates is not what it was designed for.

One clear limitation of LockWindowUpdate is that only one window can be locked for update at a time. So if two windows each try the LockWindowUpdate trick at the same time, only one of them will succeed. That's not particularly reliable, now, is it.

The purpose of LockWindowUpdate is to assist code that is drawing drag/drop feedback. If you are drawing the cursor for a drag/drop operation, you don't want the window beneath the cursor to start drawing (and thereby overwrite your beautiful cursor). So you lock the window while you draw the cursor and unlock it when the cursor leaves the window.

That's why there is only one locked window at a time: There can be only one drag/drop operation active at a time, since there is only one mouse.

Raymond Chen

**Follow**

