

Windows are not cheap objects

 devblogs.microsoft.com/oldnewthing/20050315-00

March 15, 2005



Raymond Chen

Although Windows is centered around, well, windows, a window itself is not a cheap object. What's more, the tight memory constraints of systems of 1985 forced various design decisions. Let's take for example the design of the list box control. In a modern design, you might design the list box control as accepting a list of child windows, each of which represents an entry in the list. A list box with 20,000 items would have 20,000 child windows. That would have been completely laughable in 1985. Recall that Windows was built around a 16-bit processor. Window handles were 16-bit values and internally were just near pointers into a 64K heap. A window object was 88 bytes (I counted), which means that you could squeeze in a maximum of 700 or so before you ran out of memory. What's more, menus hung out in this same 64K heap, so the actual limit was much lower. Even if the window manager internally used a heap larger than 64K (which Windows 95 did), 20,000 windows comes out to over 1.5MB. Since the 8086 had a maximum address space of 1MB, even if you devoted every single byte of memory to window objects, you'd still not have enough memory. Furthermore, making each list box item a window means that every list box would be a variable-height list box, which carries with it the complexity of managing a container with variable-height items. This goes against two general principles of API design: (1) simple things should be simple, and (2) "pay-for-play", that if you are doing the simple thing, you shouldn't have to pay the cost of the complex thing. Filling a list box with actual windows also would have made the "virtual list box" design significantly trickier. With the current design, you can say, "There are a million items" without actually having to create them. (This is also why the window space is divided into "client" and "non-client" areas rather than making the non-client area consist of little child windows.) To maintain compatibility with 16-bit Windows programs (which still run on Windows XP thanks to the WOW layer), there cannot be more than 65536 window handles in the system, because any more than that would prevent 16-bit programs from being able to talk meaningfully about windows. (Once you create your 65537'th window, there will be two windows with the same 16-bit handle value, thanks to the pigeonhole principle.) (And yes, 16/32-bit interoperability is still important even today.) With a limit of 65536 window handles, your directory with 100,000 files in it would be in serious trouble. The cost of a window object has grown over time, as new features get added to the window manager. Today it's even heftier than the svelte 88 bytes of yesteryear. It is to your advantage not to create more windows than

necessary. If your application design has you creating thousands of windows for sub-objects, you should consider moving to a windowless model, like Internet Explorer, Word, list boxes, treeview, listview, and even our scrollbar sample program. By going windowless, you shed the system overhead of a full window handle, with all the baggage that comes with it. Since window handles are visible to all processes, there is a lot of overhead associated with centrally managing the window list. If you go windowless, then the only program that can access your content is you. You don't have to worry about marshalling, cross-process synchronization, Unicode/ANSI translation, external subclassing, hooks... And you can use a gigabyte of memory to keep track of your windowless data if that's what you want, since your windowless controls don't affect any other processes. The fact that window handles are accessible to other processes imposes a practical limit on how many of them can be created without impacting the system as a whole.

I believe that WinFX uses the “everything on the screen is an element” model. It is my understanding that they've built a windowless framework so you don't have to. (I'm not sure about this, though, not being a WinFX person myself.)

Raymond Chen

Follow

