# Why does the taskbar icon for grouped windows change to something weird?

devblogs.microsoft.com/oldnewthing/20150812-00

August 12, 2015

Raymond Chen

This question came in from two customers. I'm pretty sure it was two customers rather than one customer going on a fishing expedition because the questions came six months apart, and the scenarios were different.

Suppose you remove all shortcuts to Explorer from the taskbar and the Start menu. Then you create a shortcut to Explorer and put it on the desktop. Wait, you're not done yet. Now view the Properties of that shortcut, use the *Change Icon* button, and give it some random icon. The uglier the better.

Last step: Go to the Taskbar properties and set *Taskbar buttons* to *Always combine, hide labels*.

All right, now open an Explorer window. Observe that it has the ugly icon in the taskbar rather than an icon that matches the Explorer window that you opened.

What's going on here?

Last step first: Since you configured the taskbar as *Always combine*, the icon for the Explorer does not come from the window itself, but is rather the group icon.

Okay, so where does the taskbar get the group icon from? The taskbar looks in the following places to get the group icon:

1. A shortcut on the Start menu.
2. A shortcut on the desktop.
3. The executable itself.

Normally, a shortcut is found on the Start menu, but in this case, the user explicitly removed all shortcuts to `explorer.exe` from the Start menu. That means that the winner was the shortcut on the desktop. That shortcut has a really ugly icon, so the taskbar shows the really ugly icon.

In other words, the reason you're getting an ugly icon is that when Windows tries to figure out the icon to show for Explorer groups, you deleted all the good icons and left only the ugly icon.

Okay, so why does the taskbar even bother looking at shortcuts on the Start menu and on the desktop? Why doesn't it just show the icon for the executable itself?

A lot of applications don't bother giving their executable a nice icon. The theory being, "Well, we give our Start menu shortcut a nice icon. And when the program runs, it registers a nice icon when it calls `RegisterClass`. The executable itself is buried off in the Program Files directory, which nobody should be messing with anyway, so who cares if it has an ugly icon there?" And then when the taskbar first added the "group icons" feature, a lot of programs showed the wrong icon when collapsed to a group.

3 Contoso Designer

So that's where the first rule comes from: See if there is a shortcut to the program on the Start menu. If so, then use that icon, because that's the icon the program wants to show to the user to say "Hey, run my program!"

But even with that, there were still some incorrect icons. Those were from programs who installed their shortcut on the desktop rather than the Start menu. That's why there is rule number two.

Only if there is no shortcut on the Start menu or the desktop does the taskbar look to the executable itself.

It so happens that Explorer already has to keep track of every shortcut on the Start menu and on the desktop, because it needs to keep track of any hotkeys registered by those shortcuts. Having it keep track of yet another piece of information for every shortcut wasn't too much of an extra burden.

**Bonus chatter**: Why not just create a compatibility shim for these ugly applications?

In general, when you find these sorts of compatibility issues, you can choose to fix them either by accommodating the issue in the core operating system, or by creating a compatibility shim that applies only to the applications affected by the issue. If the problem is widespread, then you just have to suck it up and put the compatibility behavior in the core operating system. Otherwise, the compatibility database would be bloated with thousands of entries. What's more, it's clear that there will be a very long tail of affected applications, seeing as the default icon for MFC applications is the generic MFC icon, and there are probably a whole lot of vertical-market and line-of-business applications that are just using

the default icon without realizing it. These applications are not big-market mainstream applications, so they will likely never come to the attention of the application compatibility team.

Raymond Chen

**Follow**