Menu item states are not reliable until they are shown because they aren't needed until then

devblogs.microsoft.com/oldnewthing/20110805-00

August 5, 2011

Raymond Chen

A question arrived from a customer (with the rather unhelpful subject line *Question for Microsoft*) wondering why, when they call `GetSystemMenu` and then ask for the states of the various menu items like `SC_MINIMIZE`, the menu item states don't reflect reality. The menu item states don't synchronize with reality until the user actually opens the system menu. There is no requirement that applications keep menu item states continuously in sync. After all, that's why we have messages like `WM_INITMENU`: To tell the application, "Whoa, we're about to show this menu, so you might want to comb its hair and pick the food out of its teeth so it can be seen by the user." Lazy evaluation is common, because maintaining states continuously can be expensive, and there's no point constantly turning items on and of and on and off if the user can't see them anyway. This is double-true for system menus, because maintaining the states continuously is not possible when the system menu is being shared across windows. The menu states are not synchronized to the window until the menu is about to be displayed. If you want to know whether the `SC_MINIMIZE` menu item *would be* enabled if the menu were shown, you can check the window styles: A window can be minimized if it has a `WS_MINIMIZEBOX` and is not already `WS_MINIMIZE`d. Similar logic can be applied to the other menu items. Well, except for `SC_CLOSE`. While in most cases the window caption determines what is enabled on the menu, the Close button works backward: It is the state of the menu item that controls whether the Close button is enabled. So in the special case of `SC_CLOSE`, you *can* query the state at any time, because for that case, the menu controls the state rather than simply reflecting it. Why is `SC_CLOSE` so special? Here come da history. The Close button was added in Windows 95. Since versions of Windows prior to Windows 95 didn't have a Close button, they didn't need a style to specify whether the Close button should be enabled or not. (You don't need a style to control something that doesn't exist.) Windows 95 added the Close button and hooked it up to the only thing that it had available, namely, the `SC_CLOSE` item on the system menu. Sure, Windows 95 could have have invented a new window style, but since `SC_CLOSE` already existed and applications were already using it, using `SC_CLOSE` to control the Close button allowed old applications to reap the benefits of the new Close button automatically. It also meant that there was one less thing you had to change when porting your program to Windows 95.

Bonus chatter: You can now answer Alex Cohn's question:

> I wonder if the EnableMenuItem method will work for minimize and maximize, too. After all, these buttons also have siblings in the Alt-space menu.

Raymond Chen

Follow