Any setting you expose to the user you implicitly expose to applications

devblogs.microsoft.com/oldnewthing/20130502-00

May 2, 2013

Raymond Chen

Often, in response to some sort of design decision, people will say, “Well, sure, you made this decision because it would allow applications to do Bad Thing, but why not expose it as a setting the user can select? For example, let the user pick a Topper Than Topmost Awesome Top Window Super Top (Extra Super edition), and keep that window on top regardless of what any application does.”

Because anything the user can do, an application can do.

Suppose there was a new context menu item for a window called *Make this Topper Than Topmost Awesome Top Window Super Top (Extra Super edition)*. Well, an application could just programmatically send the `WM_SYSCOMMAND` message with `wParam` set to `SC_TOPPERTHANTOPMOSTAWESOMETOPWINDOWSUPERTOPEXTRASUPER` .

If you say, “Nope, that context menu item is super secret and has a random command ID so nobody knows what its ID is”, well, the program could just call `GetSystemMenu` and enumerate the menu items and then extract the ID from the one whose name is *Make this Topper Than Topmost Awesome Top Window Super Top (Extra Super edition)*.

If you say, “Nope, that menu item will be hidden from enumeration, so programs which enumerate their system menu can’t see it”, well, the program could just use Accessibility to programmatically open its system menu, and then programmatically click the *Make this Topper Than Topmost Awesome Top Window Super Top Super (Extra Super edition)* button.

Anything the user can do, a program can do by simply pretending to be the user.

Raymond Chen

Follow