Why can’t you change the alignment of an edit control after it has been created?

devblogs.microsoft.com/oldnewthing/20090525-00

May 25, 2009

Raymond Chen

Commenter Kevin Eshbach asks why you cannot programatically change the text centering style of an edit control after the control has been created.

The edit control is one of the original window manager controls, so it comes with a lot of history. First thing to note is that the `WM_STYLECHANGED` message did not exist until Windows 95. The edit control predates Windows 95 by over a decade, so the short answer is “It can’t respond to the change in styles because it doesn’t even know that you changed them!”

Back in the old days, the only time a control was informed of its styles was in the `CREATESTRUCT` passed as parameters to the `WM_CREATE` and `WM_NCCREATE` messages. Consequently, most controls back in those days cached their styles in member variables so that they wouldn’t have to call `GetWindowLong(GWL_STYLE)` all the time. This was back in the days of 4.77MHz CPUs, and a call to fetch the style of a window was quite a bit bigger and slower than just reading it out of your local data segment.

Even if you decided “Forget it, I won’t bother with a cache; I’ll just call `GetWindowLong(GWL_STYLE)` to read the style each time I need it”, you open yourself up to a new problem: What if the style changes? A change in the style means that the characters in the edit box need to have their positions recalculated and the text redrawn. This is obviously not something you want to do at every single message (redrawing at every message is probably not a great idea), so you still need to cache the old style so you can avoid recalculating stuff that hasn’t changed.

Even if you say, “Okay, fine, then instead of changing the style with `SetWindowLong(GWL_STYLE)`, let’s say that if you want to change the style of an edit control, you have to use a new message, something like `EM_CHANGESTYLE`.” That way, the edit control knows when its styles have changed and doesn’t have to keep polling for them. When it gets this message, it updates the styles and recalculates all its internal state based on the new style.

Okay, sure it could've done this, but that's code that has to be written and tested, for an extremely rare usage pattern. I mean, what is the scenario where somebody needs to change an edit control from left-aligned to right-aligned on the fly? You could count all the Windows 1.0 applications on your fingers and toes (note: exaggeration), and they just decided in the dialog template which way the text should be aligned. It was determined not just at creation time; it was determined at *compile time*.

If you want to change an edit control's alignment on the fly, you can just destroy the old one and create a new one in its place with the same text.

Raymond Chen

Follow