I thought you could use SWP_FRAMECHANGED to force a control to recalculate its properties after a change in styles

devblogs.microsoft.com/oldnewthing/20140424-00

April 24, 2014

Raymond Chen

Simon Cooke dug back into his memory and asked, “Doesn’t calling `SetWindowPos` with `SWP_FRAMECHANGED` cause a recreate and re-apply of the styles?” The `SWP_FRAMECHANGED` flag does not recreate anything, but it does reapply the styles, as far as it knows. Recall that the bits in the window style break into two parts. There are the styles managed by the window manager, which are in the upper 16 bits, and there are the styles that are specific to each control, which are in the lower 16 bits. The window manager knows about the styles that it manages, but it has no clue about the styles that are specific to each control. It has no idea that the `MCIWNDF_NOPLAYBAR` style controls the toolbar in an MCI window, or that the `ES_RIGHT` style controls the alignment of text in an edit control. The `SWP_FRAMECHANGED` flag tells the window manager, “Hey, I changed some styles that affect the non-client area of the window (the window frame). Could you go and re-read those styles and apply them to the window? Thanks.” That’s sort of implied in the name: “Frame changed.”

If you want a control to re-inspect the window styles and adjust its behavior in response, you need to do something control-specific. The control might have a custom message you can send it to say, “Hey, I changed some styles that afect the client area of the window. Could you go and re-read those styles and apply them to the window? Thanks.” Or there may be special messages specifically for changing styles, such as `EM_SETREADONLY`. The fancier windows may do it automatically on receipt of the `WM_STYLECHANGED` messages.

Raymond Chen

Follow