

The per-class window styles and things really are per-class

devblogs.microsoft.com/oldnewthing/20060201-00

February 1, 2006



Raymond Chen

Earlier, I discussed [which window style bits belong to whom](#). One detail of this that I neglected to emphasize is that since the lower 16 bits of the window style are defined by the class, you can't just take styles from one class and apply them to another. For example, you can't create a button control and pass the `SS_ENDELLIPSIS` style expecting to have the text rendered with end ellipses. Because when you think you're passing `SS_ENDELLIPSIS`, you're really passing `BS_NOTIFY`:

```
#define SS_ENDELLIPSIS    0x00004000L
#define BS_NOTIFY        0x00004000L
```

The button control sees your `0x00004000L` and treats it as `BS_NOTIFY`.

Remember that at the end of the day, window styles and window messages are just numbers. If you use a per-class window style or window message, you'd better be passing it to the class that defined it.

This also applies to window extra bytes. The value returned by `GetWindowLongPtr(hwnd, DWLP_DLGPROC)` is meaningful only if `hwnd` is a dialog box. I've seen code by a major commercial software manufacturer that just runs around fiddling with the `DWLP_DLGPROC` of every window on the desktop on the assumption that "Why of course it's a dialog box, why do you ask?" Well, except that `DWLP_DLGPROC` has the numerical value of 4 (or 8 on Win64). Positive window byte indices are class-defined. Asking for `DWLP_DLGPROC` of a random window will give you the dialog procedure if that window is a dialog box, but it'll return some other internal data if the window isn't. Fortunately, most window classes don't ask for more than `sizeof(void*)` extra bytes, so the request for `DWLP_DLGPROC` just fails with an invalid parameter error. But if there happened to be a window belonging to a class with a larger number of extra bytes, that window will be in for quite a surprise when that rogue program comes in and starts messing with those extra bytes.



Raymond Chen

Follow