

Why do I have to add one when setting a class background brush to a system color?

devblogs.microsoft.com/oldnewthing/20211220-00

December 20, 2021



Raymond Chen

When you register a window class, you can specify that the background color is a system color by adding one to the system color index, and then casting the result to `HBRUSH` :

```
WNDCLASS wc;  
...  
wc.hbrBackground = (HBRUSH)(COLOR_WINDOW + 1);
```

Why do we add one?

To make sure that the result is not zero.

The `COLOR_SCROLLBAR` system color is index zero. If we didn't add one to the system color indices, then an attempt to set the background color to the scroll bar color would end up with `(HBRUSH)0` , which is a null pointer. When the system saw that you set `hbrBackground` to a null pointer, it wouldn't know whether you meant the window to have no background color (null pointer), or whether you wanted it to have the system scroll bar color (`COLOR_SCROLLBAR`).

In other words, adding one ensures that the space of “system colors smuggled inside a brush handle” does not overlap with the space of regular brush handles.

In retrospect, this was one of those “too clever” hacks born out of the days of 16-bit Windows and systems with only 256KB of memory. Nowadays, we would say, “Just set the background to `GetSysColorBrush(COLOR_WINDOW)` .” But back then, `GetSysColorBrush(COLOR_WINDOW)` didn't exist.

[Raymond Chen](#)

Follow

