Why are many Windows user interface elements positioned at multiples of 4 or 8 pixels?

devblogs.microsoft.com/oldnewthing/20221025-00

October 25, 2022

Raymond Chen

Some time ago, we learned that Windows 95 positioned windows at multiples of 8 pixels in order to make bit block transfers more efficient. Is that why many Windows user interface elements are still positioned at multiples of 4 and 8 pixels? (And why four as well as eight?)

No, it's not about bit block transfer efficiency.

Oh, do designers just have this compulsion about multiples of four and eight, to the point where they can glance at a screen, and if something is not at a multiple of four pixels, they get this odd feeling that the screen is somehow *wrong* and needs to be *fixed*?

No, that's not it either. Or at least I don't think that's true.

The reason for positioning elements at multiples of four or eight comes down to screen scaling.

As the screen pixel density increases, Windows applies a scaling factor so that objects on the screen appear at roughly the same visual size. If you have a 192 pixels-per-inch display, it will show a rectangle at the same physical size as the same rectangle on a 96 pixels-per-inch display, but with four times as many pixels (twice horizontally and twice vertically).

I say approximately, because Windows doesn't go down to the last pixel to match sizes. If your screen pixel density is 193 pixels per inch, Windows will treat it as 192 pixels per inch for scaling purposes. If it had used the exact value of 193 pixels per inch, then all of your screen elements would be scaled by 201%, and that extra 1% is going to make things blurry: A box that started out as 100 pixels wide will end up scaled to 201 pixels wide, and that means that almost none of the source pixels land on an exact pixel boundary when scaled up.

The missing piece of the puzzle is the fact that Windows recommends scaling factors in multiples of 25%: 100%, 125%, 150%, 175%, 200%, 225%, 250%, 300%, 350%, 400%, 450%, 500%.[1]

Now it all comes together.

If you arrange for all of your unscaled pixel coordinates to be multiples of 4 (or, for added safety, multiples of 8), then after scaling, they will still be exact integers.

[1] If you look at the `DEVICE_SCALE_FACTOR` enumeration, you'll see values that aren't multiples of 25%. Those are old values left over from Windows 8 and Windows Phone. They're not used any more.

Raymond Chen

Follow