What is the deal with the SM_CXCURSOR system metric?

devblogs.microsoft.com/oldnewthing/20210819-00

Raymond Chen

There is a system metric known as `SM_CXCURSOR`. It ostensibly tells you the width of the cursor, but that doesn't really hold up to scrutiny. First of all, there are multiple cursors in the system; which one does it give you the width of? And if you try it, you find that even if the cursors are all the same size, the metric doesn't give you the right number. What's going on?

Rewind to 16-bit Windows.

In 16-bit Windows, the standard cursors were provided by the display driver. All of those cursors were the same size, so a single metric was sufficient.

Windows 95 added the ability to customize the standard cursors, and that's when things started to get messy. With this new feature, you could customize your cursors to be a size different from the size the display driver would have given. The `SM_CXCURSOR` metric still gave you the size of the driver-provided cursors, though, even if they weren't the cursors that were in active use.

In Windows NT, the story is different. The display drivers are no longer in the default cursor business. The default cursors come from the window manager, but most users probably choose a custom one from the Mouse control panel.

Okay, so what does `SM_CXCURSOR` mean in this new world where nobody is using cursors from the display driver?

The `SM_CXCURSOR` metric returns the width of the cursor if the system were using the defaults. Which it rarely does any more. You can use them as the "nominal" cursor size, but understand that the actual cursors may not be that size. So what size are they, really? We'll look into that next time.

Bonus chatter: In case you were wondering, here are the nominal cursor sizes, as of this writing. These values are not contractual and can change at any time, but I'm providing them to satisfy your curiosity:

DPI (scale)					Cursor size	
	x	<	144dpi (150%)		32 ×	32 (100%)
144dpi (150%)	≤	x	<	192dpi (200%)	48 ×	48 (150%)
192dpi (200%)	≤	x	<	288dpi (300%)	64 ×	64 (200%)
288dpi (300%)	≤	x	<	384dpi (400%)	96 ×	96 (300%)
384dpi (400%)	≤	x			128 × 128 (400%)	

Raymond Chen

Follow