What is the logic behind the thumb size and position calculation in scroll bars?

devblogs.microsoft.com/oldnewthing/20090921-00

September 21, 2009

Raymond Chen

Commenter sarathc asks, “How do we implement a custom scroll bar as Windows does? What is the logic behind the thumb size and position calculation? How we could dynamically manage it?” Let’s look at the three questions in turn. To implement a custom scroll bar… don’t do it. It’s just not worth the effort, and there will almost always be little seams, like not lighting up when the mouse hovers over them. The logic behind the thumb size and position calculation I thought I covered in my scroll bar series. The size of the thumb relative to the size of the scroll bar is the same as the page size relative to the scroll bar range. In other words:

> thumb size / scroll bar size = page size / scroll bar range

A little high school algebra tells you, then, that

> thumb size = scroll bar size * page size / scroll bar range

There may be some off-by-one errors in the above formula, and some special tweaks for extreme cases (you don’t want a thumb smaller than one pixel after all), but that’s the basic idea. Similarly, the screen position of the thumb relative to the scroll bar is equal to the programmatic thumb position relative to the scroll bar range (roughly).

To dynamically manage it, use the usual scroll bar functions like `SetScrollInfo`.

Raymond Chen

Follow