

Why are RECTs endpoint-exclusive?

 devblogs.microsoft.com/oldnewthing/20040218-00

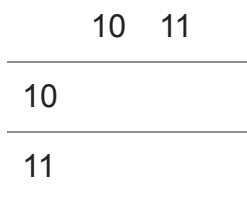
February 18, 2004



Raymond Chen

Endpoint-exclusive RECTs and lines are much easier to work with. For example, the width of a rectangle is `right - left`, and its height is `bottom - top`. If rectangles were endpoint-inclusive, then there would be annoying `+1`'s everywhere. End-point exclusive rectangles also scale properly. For example, suppose you have two rectangles `(0,0)-(100,100)` and `(100,100)-(200,200)`. These two rectangles barely touch at the corner. Now suppose you magnify these rectangles by 2, so they are now `(0,0)-(200,200)` and `(200,200)-(400,400)`. Notice that they still barely touch at the corner. Also the length of each edge doubled from 100 to 200. Now suppose endpoints were inclusive, so the two rectangles would be `(0,0)-(99,99)` and `(100,100)-(199,199)`. Now when you double them, you get `(0,0)-(198,198)` and `(200,200)-(398,398)`. Notice that they no longer touch any more because `(199,199)` is missing. Note also that the length of the side of the square is now 199 pixels instead of 200. Similar problems occur if you need to do subpixel computations. "But that's silly — who ever does magnification or subpixel computations?" Well, magnification is used more than you think. In addition to the obvious things like zooming in/out, it's also used in printing (since printers are 300dpi but the screen is usually much lower resolution) and in GDI mapping (`ScaleWindowExtEx`, `StretchBlt`). And subpixel computations are used in anti-aliasing.

With apologies to Alvy Ray, I think the best way to interpret this is to view pixels as living **between** coordinates, not at them. For example, here's a picture of the pixel that lives between `(10,10)` and `(11,11)`. (In other words, this pixel **is** the rectangle `(10,10)-(11,11)`.)



With this interpretation, the exclusion of the endpoint becomes much more natural. For example, here's the rectangle `(10,10)-(13,12)`:

10 11 12 13

10

11

12

Observe that this rectangle starts at (10,10) and ends at (13,12), just like its coordinates say.

Raymond Chen

Follow

