

The format rectangle is recalculated whenever the window resizes, and that's a good thing

 devblogs.microsoft.com/oldnewthing/20090817-00

August 17, 2009



Raymond Chen

Reader [asmguru62](#) asks why a custom format rectangle of a multi-line edit control is lost when the edit control is resized. That's right, the format rectangle is recalculated whenever the window resizes, and that's a good thing. Imagine if it weren't recalculated. You create a multi-line edit control. Like many programs, you might create it at a temporary size while you try to figure out what its real size should be. You might even resize the control dynamically as the container is resized. (After all, everybody wants resizable dialogs.) What you definitely don't want is the format rectangle of the edit control to be locked to the size the window was originally created at. "Well, yeah," [asmguru62](#) says, "sure that may be desirable if the application is using the default margins, but if the margins have been customized, then shouldn't those margins be preserved?" First, who says that your call to `EM_SETRECT` was to established fixed-size margins? All the function knows is that you said "Put the text inside this rectangle." It has no idea how you computed that rectangle. Did you subtract a fixed number of pixels? Did you use a percentage? Did you set the rectangle to force the text into a rectangle whose width and height form the golden mean? Second, if you want to set the margins, then set the margins. The `EM_SETMARGINS` message lets you specify the size of the left and right margins of a multi-line edit control. The edit control will take your margins into account when it recalculates its format rectangle after a resize. In other words, `EM_SETRECT` is the low-level function that lets you manipulate the edit control's internal formatting rectangle, the same rectangle that the edit control itself manipulates in response to the things that edit controls naturally respond to. There is no fancy inference engine here that says, "Let me attempt to reverse-engineer how this rectangle is related to the client rectangle so I can carry forward this computation when I recalculate the format rectangle." Think of it as reaching in and directly whacking one of the edit control's private member variables.

If you want something higher-level, then use `EM_SETMARGINS` .

[Raymond Chen](#)

Follow



