

When I define a window class with no default cursor, what is the explanation for the cursors that appear in my client area?

 devblogs.microsoft.com/oldnewthing/20250424-00/?p=111114

April 24, 2025



A customer created a window class with no default cursor, meaning that they set the `hCursor` to `nullptr`. What they saw was that the window showed up with an hourglass cursor, and that cursor never went away on its own. But if the user moved the mouse over a border, then the border arrow cursor carried into the client area. What's going on?

What's going on is that by setting the cursor to `nullptr`, you're saying "I will take care of the cursor." Some time ago, I discussed [how the cursor gets set](#), and if nobody else takes responsibility, then `DefWindowProc` sets the cursor to the window class's registered class cursor.

If there is no registered class cursor (if you set it to `nullptr`), then that final fallback step doesn't do anything, and nobody sets the cursor.

If nobody sets the cursor, then the cursor remains unchanged, and whatever cursor is currently set continues to be the cursor. The cursor remains set until somebody else sets a new one. If the cursor wanders over a border, then the `DefWindowProc` function sets the border cursor. And then when it wanders into the client area, nobody sets the cursor, so the old border cursor remains.

Nobody actively asked for the border cursor to appear in the client area. Rather, nobody said what they wanted to appear in the client area, so nothing was changed.

The moral of the story is that if you set your class cursor to `nullptr`, then you are assuming responsibility for handling the `WM_SETCURSOR` message and making sure that, eventually, somehow, a cursor gets set. If you fail to fulfill that obligation, then nobody sets the cursor, and you get leftover garbage.