

# The cursor isn't associated with a window or a window class; it's associated with a thread group

[devblogs.microsoft.com/oldnewthing/20110207-00](http://devblogs.microsoft.com/oldnewthing/20110207-00)

February 7, 2011



Raymond Chen

In my earlier discussion of the fact that changing a class property affects all windows of that class, commenters [LittleHelper](#) and [Norman Diamond](#) wanted to know “Why is the cursor associated with class and not a window?” This is another one of those questions that start off with an invalid assumption. The cursor is not associated with a class. The cursor is not associated with a window. The cursor is associated with an *input state*. (Initially, each thread has its own input state, but functions like `AttachThreadInput` can cause threads to share their input states.) As we saw when we explored the process by which the cursor gets set, the cursor-setting process is initiated by the `WM_SETCURSOR` message, which is percolated up and down the window hierarchy until somebody calls `SetCursor` and returns `TRUE` to say “Okay, I set the cursor. You can stop searching now.” And that cursor remains in effect until somebody else in the same thread group calls `SetCursor`. It so happens that the `DefWindowProc` function, when asked to set a cursor, will use the window’s class cursor. But that’s just the default in the absence of any customization to the contrary. If you want to customize the cursor when it is over a particular window, then use the customization; don’t go changing the default. If you change the default, then you affect what happens to all the other windows of the class. Just handle the `WM_SETCURSOR` message to establish your “per-window cursor”. (And you can be even more specific than just per-window. For example, you might decide to show a hand cursor if the user is over a hyperlink but an arrow cursor otherwise.)

Many of the fields in the `WNDCLASS` structure are merely defaults which are applied to windows of the class. You can still override them on a per-window basis.

Field	How to override
<code>lpfnWndProc</code>	<code>SetWindowLongPtr(GWLP_WNDPROC)</code>
<code>hIcon</code>	<code>SendMessage(WM_SETICON)</code>
<code>hCursor</code>	Handle the <code>WM_SETCURSOR</code> message

<code>hbrBackground</code>	Handle the <code>WM_ERASEBKGD</code> message
<code>lpszMenuName</code>	<code>SetMenu()</code>

(This is the same table I wrote up some time ago, but the original table didn't have an entry for the window procedure, so this table is *slightly* more complete.)

Raymond Chen

**Follow**

