Why doesn’t my custom-drawn trackbar get a paint notification when the position changes from 1 to 0?

devblogs.microsoft.com/oldnewthing/20161005-00

October 5, 2016

Raymond Chen

A customer reported that under certain conditions, their custom-drawn trackbar does not generate a `NM_CUSTOMDRAW` message.

We have found that the trackbar control in the shell common controls library does not generate a `NM_CUSTOMDRAW` message when the position changes from 1 to 0 and the trackbar's range is sufficiently high.

We start with the trackbar position at 1.

−1000 +1000

| | |

Current value: 1

And then we send the `TBM_SETPOS` message to set the trackbar position to zero. The result is this:

−1000 +1000

| | |

Current value: 1

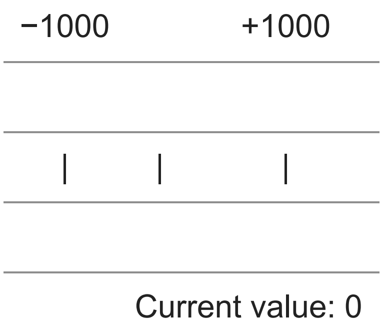
Observe that the "Current value" is reported as 1, even though we changed the value to 0. On the other hand, if we start with the position at −1:

−1000 +1000

| | |

Current value: −1

then when we send the `TBM_SETPOS` message to change the position to zero, we do get a `NM_CUSTOMDRAW` message, and the "Current value" updates.

> 
>
>
> We have been able to reproduce this problem on every version of the trackbar as far back as we tested.

Everything is working as it should.

The `NM_CUSTOMDRAW` notification lets you customize how the common control draws itself. If there is nothing that needs to be redrawn, then there is no `WM_PAINT` message and consequently no `NM_CUSTOMDRAW` notification.

When the trackbar range is large, then multiple positions have the same visual appearance. This is a natural consequence of the pigeonhole principle: There are 500 (say) pixel positions that the thumb could be drawn, but there are 2001 possible positions, so around four thumb positions all correspond to the same visual appearance.

What appears to be happening is that positions 0 and 1 share the same visual appearance, so when the thumb position changes between 0 and 1, there is no visual change and therefore no `NM_CUSTOMDRAW` message.

On the other hand, it appears that positions −1 and 0 have different visual apperances, which is why you get a `NM_CUSTOMDRAW` message when the position changes from −1 to 0.

It sounds like the application is using the `NM_CUSTOMDRAW` notification to detect when the trackbar position has changed. That's not what it's for. The `NM_CUSTOMDRAW` notification is for letting you customize the way the trackbar is drawn.

If you want to know when the trackbar position changes, listen for the `WM_HSCROLL` message. Note, however, that the `WM_HSCROLL` message is not generated if the program itself changes the position via the `TBM_SETPOS` message, on the theory that since the program itself changed the value, it can update its own state right there. No need to tell the program what it already knows.

Bonus chatter: Not generating a notification for program-generated position changes also helps avoid infinite loops. After the program changes the trackbar position, it receives the change notification, and in response to the notification, the program tries to update some state. But the state update fails, so the program tries to undo the change and set the position back. This reset generates its own change notification, and the program responds to the notification by trying to update that same state (to the old value), which still fails, so the program tries to undo the change and set the position back, which generates yet another change notification, and so on.

The theory here is that the code that is listening for the `WM_HSCROLL` or `WM_HSCROLL` message is also the code that is sending the `TBM_SETPOS` message, so there’s no point in telling the program something it already knows.

Exercise: Suppose you have a trackbar, and you want to let anybody send it a `TBM_SETPOS` message to change the trackbar position, but you also want to be notified when that happens. How would you do that?

Raymond Chen

Follow