Scrollbars part 9 – Maintaining the metaphor

devblogs.microsoft.com/oldnewthing/20030909-00

September 9, 2003

Raymond Chen

When a document is displayed with scrollbars, the metaphor is that the window is a viewport onto the entire document, only a portion of which is visible at the moment. The default behavior of a resize, however, is to maintain the origin at the upper left corner of the client area, which breaks the metaphor when the window is resized at the top or left edge.

Suppose, for example, that the top line in the document is line ten. If the user grabs the top edge of the window and resizes upwards by one line (in an attempt to view line nine), the default behavior is to maintain the origin, which keeps line ten at the top of the window. The visual effect is that the window has scrolled upwards one line, subverting the user's attempt to view line nine.

This is one of the subtleties of scrollbars which users rarely consciously notice, but when it doesn't work, it gives the impression that computers never quite get things right.

Let's fix our scrolling behavior to maintain the viewport metaphor. We'll do it in several steps. First, we'll over-preserve the metaphor. Add the following new section to *OnWindowPosChanging*:

```
 BOOL OnWindowPosChanging(HWND hwnd, LPWINDOWPOS pwp)
{
    if (!(pwp->flags & SWP_NOMOVE)) {
        RECT rc;
        GetWindowRect(hwnd, &rc);
        int dy = pwp->y - rc.top;
        ScrollDelta(hwnd, dy / g_cyLine);
    }

     if (!(pwp->flags & SWP_NOSIZE)) {
        RECT rc = { 0, 0, pwp->cx, pwp->cy };
        AdjustSizeRectangle(hwnd, WMSZ_BOTTOM, &rc);
        pwp->cy = rc.bottom;
    }
    return 0;
}
```

Now run the program, move the scrollbar thumb to somewhere in the middle, and resize the top edge of the window upwards and downwards. Notice that the existing lines on the screen don't move; all that resizing the top of the window does is expose or hide lines at the top. You already experience this behavior when resizing the bottom edge; now you get it at the top, too.

There are several things wrong with this code, however.

First, observe that it is trying too hard. Grab the window and move it across the screen. Observe that it still tries to preserve the aperture metaphor. (Even worse: It depends on how fast you move your mouse. The effect is more noticeable if you disable "Show window contents while dragging".) This is probably undesirable.

Second, notice all the horrible flicker.

We'll address these two problems in turn.

Fixing the overzealousness is the easier problem. First, we do the work only if the window is simultaneously moving **and** sizing. This prevents simple moving from triggering the metaphor behavior.

```
 BOOL OnWindowPosChanging(HWND hwnd, LPWINDOWPOS pwp)
{
    if (!(pwp->flags & SWP_NOSIZE)) {


        RECT rc = { 0, 0, pwp->cx, pwp->cy };
        AdjustSizeRectangle(hwnd, WMSZ_BOTTOM, &rc);
        pwp->cy = rc.bottom;


         if (!(pwp->flags & SWP_NOMOVE)) {
            RECT rc;
            GetWindowRect(hwnd, &rc);
            int dy = pwp->y - rc.top;
            ScrollDelta(hwnd, dy / g_cyLine);
        }
    }
    return 0;
}
```

Now if you grab the window and move it around, we don't do the metaphor thing because the size didn't change.

However, if you maximize a window, the metaphor code kicks in. But that's easy to fix: Only do the metaphor if the top edge changes and the bottom edge doesn't.

```
 BOOL OnWindowPosChanging(HWND hwnd, LPWINDOWPOS pwp)
{
    if (!(pwp->flags & SWP_NOSIZE)) {
        RECT rc = { 0, 0, pwp->cx, pwp->cy };
        AdjustSizeRectangle(hwnd, WMSZ_BOTTOM, &rc);
        pwp->cy = rc.bottom;

         if (!(pwp->flags & SWP_NOMOVE)) {
            RECT rc;
            GetWindowRect(hwnd, &rc);
            if (rc.bottom == pwp->y + pwp->cy) {
                int dy = pwp->y - rc.top;
                ScrollDelta(hwnd, dy / g_cyLine);
            }
        }
    }
    return 0;
}
```

We still have a problem with the flicker, though. Before we can fix that, we will need a deeper understanding of the `WM_NCCALCSIZE` message.

Raymond Chen

Follow