Scrollbars bart 6 – The wheel

devblogs.microsoft.com/oldnewthing/20030807-00

August 7, 2003

Raymond Chen

The mouse wheel is tricky because the mouse wheel UI guidelines indicate that you should scroll by a user-specified amount for each "click" of the mouse, where one click is `WHEEL_DELTA` mouse units (called a "detent"). There are two subtle points about the above requirement: First, that the amount of scrolling is a user setting which must be respected, and second, that the wheel can report values that are not perfect multiples of `WHEEL_DELTA`.

In particular, there is the possibility that a high-resolution mouse wheel will report wheel scroll units smaller than `WHEEL_DELTA`. For example, consider a wheel mouse that supports "half-clicks". When you turn the wheel halfway between clicks, it reports `WHEEL_DELTA/2`, and when you reach a full click, it reports another `WHEEL_DELTA/2`. To handle this properly, you need to make sure that by the time the full click is reached, the window has scrolled by exactly the amount it would have scrolled if the user had been using a low-resolution wheel that reported a single wheel motion of `WHEEL_DELTA`.

(I once referred to this in email as a "sub-detent wheel" and was accused of coining a phrase.)

To handle the first point, we requery the user's desired scroll delta at each mouse wheel message. To handle the second point, we accumulate detents as they arrive and consume as many of them as possible, leaving the extras for the next wheel message.

```
int g_iWheelCarryover;      /* Unused wheel ticks */
LRESULT OnMouseWheel(HWND hwnd, int xPos, int yPos, int zDelta, UINT fwKeys)
{
    UINT uScroll;
    if (!SystemParametersInfo(SPI_GETWHEELSCROLLLINES, 0, &uScroll, 0)) {
        uScroll = 3;    /* default value */
    }
    /*
     *  If user specified scrolling by pages, do so.
     */
    if (uScroll == WHEEL_PAGESCROLL)
    {
        uScroll = g_cLinesPerPage;
    }
    /*
     *  If user specified no wheel scrolling, then don't do wheel scrolling.
     *  This also avoids a divide-by-zero below.
     */
    if (uScroll == 0)
    {
        return 0;
    }
    zDelta += g_iWheelCarryover;    /* Accumulate wheel motion */
    /*
     *  See how many lines we should scroll.
     *  This relies on round-towards-zero.
     */
    int dLines = zDelta * (int)uScroll / WHEEL_DELTA;
    /*
     *  Record the unused portion as the next carryover.
     */
    g_iWheelCarryover = zDelta - dLines * WHEEL_DELTA / (int)uScroll;
    /*
     *  Do the scrolling.
     */
    ScrollDelta(hwnd, -dLines);
    return 0;
}
    /* Add to WndProc */
    HANDLE_MSG(hwnd, WM_MOUSEWHEEL, OnMouseWheel);
```

Exercise: What is the significance of the `(int)` cast in the computation of `dLines` ?

Exercise: Assuming you don't have a high-resolution wheel mouse, how would you test that your sub-detent mouse wheel handling was working properly?

Raymond Chen

Follow