

Scrollbars part 10 – Towards a deeper understanding of the WM_NCCALCSIZE message

 devblogs.microsoft.com/oldnewthing/20030911-00

September 11, 2003



Raymond Chen

When your window is resized, Windows uses the `WM_NCCALCSIZE` message to determine where your window's client area lives inside your window rectangle.

There are two forms of the `WM_NCCALCSIZE` message. The simple form merely takes a window rectangle and returns a client rectangle. This is useful for resizing a window to have a desired client rectangle, taking menu wrapping into account. The `AdjustWindowRectEx` function cannot take menu wrapping into account because it doesn't know which menu you are using. (Notice that there is no `HMENU` or `HWND` parameter to `AdjustWindowRectEx`.)

```

void
SetWindowClientSize(HWND hwnd, int cx, int cy)
{
    HMENU hmenu = GetMenu(hwnd);
    RECT rcWindow = { 0, 0, cx, cy };
    /*
     * First convert the client rectangle to a window rectangle the
     * menu-wrap-agnostic way.
     */
    AdjustWindowRectEx(&rcWindow, GetWindowStyle(hwnd), hmenu != NULL,
                      GetWindowExStyle(hwnd));
    /*
     * If there is a menu, then check how much wrapping occurs
     * when we set a window to the width specified by AdjustWindowRect
     * and an infinite amount of height. An infinite height allows
     * us to see every single menu wrap.
     */
    if (hmenu) {
        RECT rcTemp = rcWindow;
        rcTemp.bottom = 0x7FFF; /* "Infinite" height */
        SendMessage(hwnd, WM_NCCALCSIZE, FALSE, (LPARAM)&rcTemp);
        /*
         * Adjust our previous calculation to compensate for menu
         * wrapping.
         */
        rcWindow.bottom += rcTemp.top;
    }
    SetWindowPos(hwnd, NULL, 0, 0, rcWindow.right - rcWindow.left,
                  rcWindow.bottom - rcWindow.top, SWP_NOMOVE | SWP_NOZORDER);
}

```

Exercise: Explain why we used 0x7FFF to represent infinite height.

Exercise: Explain the line `rcWindow.bottom += rcTemp.top`.

Raymond Chen

Follow

