

Scrollbars, part 2

 devblogs.microsoft.com/oldnewthing/20030725-00

July 25, 2003



Raymond Chen

Managing scrollbars is not hard in principle. The basic idea is not that difficult, but there are a lot of small details that need to be ironed out. If you don't get the details just right, your program will feel odd in a strange way that you often can't pinpoint, much like a subtle background hum that makes you feel uneasy without realizing it. Getting the details right is important to making your program feel crisp and clean.

The base program

Let's start with a basic program and gradually add scrollbar features to it. The basic program merely displays one hundred numbered lines. Add these variables to the scratch program:

```
HFONT g_hfList;          /* Font for list */
int g_cyLine;           /* Height of each line */
int g_cItems = 100;    /* Number of items */
```

and add these functions to the scratch program:

```

BOOL
OnCreate(HWND hwnd, LPCREATESTRUCT lpcs)
{
    /* Create the font we use for the list */
    LOGFONT lf;
    SystemParametersInfo(SPI_GETICONTITLELOGFONT, sizeof(lf), &lf, 0);
    g_hfList = CreateFontIndirect(&lf);
    if (!g_hfList) return 0;
    /* Compute the height for each line */
    HDC hdc = GetDC(hwnd);
    HFONT hfPrev = SelectFont(hdc, g_hfList);
    SelectFont(hdc, hfPrev);
    SIZE siz;
    GetTextExtentPoint(hdc, TEXT("0"), 1, &siz);
    g_cyLine = siz.cy;
    ReleaseDC(hwnd, hdc);
    return 1;
}

void
OnDestroy(HWND hwnd)
{
    DeleteObject(g_hfList);
    PostQuitMessage(0);
}

/* This is a separate function for now; you'll see why later */
void
PaintSimpleContent(HWND hwnd, PAINTSTRUCT *pps)
{
    HFONT hfPrev = SelectFont(pps->hdc, g_hfList); /* Use the right font */
    for (int i = 0; i < g_cItems; i++) {           /* Print all the lines */
        char szLine[256];
        int cch = wsprintf(szLine, "This is line %d", i);
        TextOut(pps->hdc, 0, i * g_cyLine, szLine, cch);
    }
    SelectFont(pps->hdc, hfPrev);
}

void
PaintContent(HWND hwnd, PAINTSTRUCT *pps)
{
    PaintSimpleContent(hwnd, pps);
}

```

We now have a base program upon which to build.

I'm sorry this is taking so long to get off the ground. But things finally get interesting in part 4, honest.

Raymond Chen

Follow

