Scrollbars, part 3: Optimizing the paint cycle

devblogs.microsoft.com/oldnewthing/20030729-00

July 29, 2003

Raymond Chen

Observe that we paint all 100 lines in our paint handler, even though most of them aren't visible. This is a problem if there are a large number of items, or if painting an item is time-consuming.

So instead, we optimize our paint cycle so as to paint only the elements which intersect the paint rectangle.

```
void
PaintSimpleContent(HWND hwnd, PAINTSTRUCT *pps)
{
    HFONT hfPrev = SelectFont(pps->hdc, g_hfList);  /* Use the right font */
    int iMin = max(pps->rcPaint.top / g_cyLine, 0);
    int iMax = min((pps->rcPaint.bottom + g_cyLine - 1) / g_cyLine, g_cItems);
    for (int i = iMin; i < iMax; i++) {
        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);
}
```

Exercise: Explain the formulas for *iMin* and *iMax*. Explain why the seemingly equivalent formula

```
    int iMax = min((pps->rcPaint.bottom - 1) / g_cyLine + 1, g_cItems);
```

is wrong. Then explain why it doesn't really matter too much.

Raymond Chen

Follow