

# A program for my nieces: The ABCs, part 1

---

 [devblogs.microsoft.com/oldnewthing/20130708-00](http://devblogs.microsoft.com/oldnewthing/20130708-00)

July 8, 2013



Raymond Chen

I'm going to spend the next few weeks developing a Little Program in several parts. This is a program I wrote for my nieces, who always wanted to play with my laptop (instead of playing with *me*).

Initially, I fired up Notepad and maximized it, and cranked the font size, but that became cumbersome, because I had to reset the font size and Word Wrap setting when they were done. On top of that, my eldest niece complained that some of the the letters were “wrong”: The shape of the capital J in the font that I use does not match the shape of the capital J that my niece was taught. (The top serif didn't match.)

Having to change the font and then reset it was enough to make me decide to write my own program for my nieces to play with. I started with [the scratch program](#) and made these changes:

```
HFONT g_hfEdit;
#define MARGIN 20
void
OnSize(HWND hwnd, UINT state, int cx, int cy)
{
    if (g_hwndChild) {
        MoveWindow(g_hwndChild,
                  MARGIN, MARGIN,
                  cx - 2 * MARGIN,
                  cy - 2 * MARGIN, TRUE);
    }
}
```

The `MARGIN` puts a little space around the edit control so it doesn't jam up against the edges of the screen.

```

void
OnNcDestroy(HWND hwnd)
{
    if (g_hfEdit) DeleteObject(g_hfEdit);
    PostQuitMessage(0);
}
// HANDLE_MSG(hwnd, WM_DESTROY, OnDestroy);
HANDLE_MSG(hwnd, WM_NCDESTROY, OnNcDestroy);

```

The cleanup of the font is done in the `WM_NCDESTROY` handler because that runs after the child windows have been destroyed. That way, we don't destroy a font while the edit control is still using it.

```

BOOL
OnCreate(HWND hwnd, LPCREATESTRUCT lpcs)
{
    g_hfEdit = CreateFont(-72, 0, 0, 0, FW_NORMAL,
                        FALSE, FALSE, FALSE, DEFAULT_CHARSET,
                        OUT_DEFAULT_PRECIS,
                        CLIP_DEFAULT_PRECIS,
                        DEFAULT_QUALITY,
                        DEFAULT_PITCH,
                        TEXT("Miriam"));
    if (!g_hfEdit) return FALSE;
    g_hwndChild = CreateWindow(
        TEXT("edit"),           /* Class Name */
        NULL,                   /* Title */
        WS_CHILD | WS_VISIBLE |
        ES_UPPERCASE | ES_MULTILINE, /* Style */
        0, 0, 0, 0,             /* Position and size */
        hwnd,                   /* Parent */
        NULL,                   /* No menu */
        g_hinst,                /* Instance */
        0);                     /* No special parameters */
    if (!g_hwndChild) return FALSE;
    SetWindowFont(g_hwndChild, g_hfEdit, TRUE);
    return TRUE;
}

```

When our main window is created, we create our helper edit control. It is a multi-line edit control without any scroll bars that forces its contents to uppercase, since they haven't learned lowercase letters yet.

My program doesn't do any painting, so I deleted the `WM_PAINT` and `WM_PRINTCLIENT` handlers.

On the other hand, it needs to transfer focus to the edit control, so that switching to the application puts you in typing mode immediately:

```

void OnSetFocus(HWND hwnd, HWND hwndOldFocus)
{
    if (g_hwndChild) {
        SetFocus(g_hwndChild);
    }
}
HANDLE_MSG(hwnd, WM_SETFOCUS, OnSetFocus);

```

Finally, I create the window as a fullscreen popup, so that all my nieces get is a clean screen with no window chrome. (I was using the new Microsoft design language before it was cool.)

```

hwnd = CreateWindow(
    TEXT("Scratch"),           /* Class Name */
    TEXT("ABC"),              /* Title */
    WS_POPUP,                 /* Style */
    0, 0,                     /* Position */
    GetSystemMetrics(SM_CXSCREEN),
    GetSystemMetrics(SM_CYSCREEN), /* Size */
    NULL,                     /* Parent */
    NULL,                     /* No menu */
    hinst,                    /* Instance */
    0);                       /* No special parameters */

```

And there we have it. A simple program with an edit control that my nieces can use for typing.

They call this program ABC. Now when I go over to their house, they ask, “Can I play ABC?”

This program served well for a first pass, but my nieces naturally discovered problems with it. We’ll look at them in future weeks.

Remember, since this is a Little Program, I’m skipping a lot of error checking, and I’m assuming that the system has only one monitor (because it runs on my laptop).

Raymond Chen

**Follow**

