

How do I print the contents of a rich text control?

 devblogs.microsoft.com/oldnewthing/20070112-02

January 12, 2007



Raymond Chen

For some reason, people are really puzzled by rich edit printing. I'm no expert on printing, but even I was able to figure it out. The kernel is the `EM_FORMATRANGE` message. Each time you call it, a little bit more of the rich text control is printed, and the message returns the index of the first unprinted character, which you can pass back in to print the next chunk.

The rest is just setting up and tearing down.

```
BOOL PrintRTF(HWND hwnd, HDC hdc)
{
    int cxPhysOffset = GetDeviceCaps(hdc, PHYSICALOFFSETX);
    int cyPhysOffset = GetDeviceCaps(hdc, PHYSICALOFFSETY);
    int cxPhys = GetDeviceCaps(hdc, PHYSICALWIDTH);
    int cyPhys = GetDeviceCaps(hdc, PHYSICALHEIGHT);
    SendMessage(hwnd, EM_SETTARGETDEVICE, (WPARAM)hdc, cxPhys);
    FORMATRANGE fr;
    fr.hdc = hdc;
    fr.hdcTarget = hdc;
    fr.rc.left = cxPhysOffset;
    fr.rc.right = cxPhysOffset + cxPhys;
    fr.rc.top = cyPhysOffset;
    fr.rc.bottom = cyPhysOffset + cyPhys;
    SendMessage(hwnd, EM_SETSEL, 0, (LPARAM)-1);
    SendMessage(hwnd, EM_EXGETSEL, 0, (LPARAM)&fr.chrg);
    BOOL fSuccess = TRUE;
    while (fr.chrg.cpMin < fr.chrg.cpMax && fSuccess) {
        fSuccess = StartPage(hdc) > 0;
        if (!fSuccess) break;
        int cpMin = SendMessage(hwnd, EM_FORMATRANGE, TRUE, (LPARAM)&fr);
        if (cpMin <= fr.chrg.cpMin) {
            fSuccess = FALSE;
            break;
        }
        fr.chrg.cpMin = cpMin;
        fSuccess = EndPage(hdc) > 0;
    }
    SendMessage(hwnd, EM_FORMATRANGE, FALSE, 0);
    return fSuccess;
}
```

We start by getting the dimensions of the page and telling the rich edit control what we intend to render to by using the `EM_SETTARGETDEVICE` message. Next, we need to fill out our `FORMATRANGE`, which we do by specifying the `HDC` we are rendering to, as well as the paper dimensions. But what about the character range? We are lazy and let the rich edit control take care of it for us: We select all the text and then ask the rich edit control to tell us what we just selected, which comes back in the form of a `CHARRANGE`, which is exactly what we needed.

Next comes the printing loop. While there is still text to print (and we haven't encountered an error), we start a new page, ask the rich edit control to render that page, remember where the next page should begin, and end the current page. There's a little sanity check in there to make sure that the rich edit control made forward progress; if not, then we'll end up in an infinite loop spewing out blank pages! (I have no idea whether this is theoretically possible, but I'm going to protect against it just the same.)

Once the printing loop is complete, we clean up by sending one last `EM_FORMATRANGE` message to tell the rich edit control that we're all done and it can discard the information it cached.

We can take all the information we've learned over the past few days to make a simple "print RTF" program.

```

int CALLBACK
_tWinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
          LPTSTR lpCmdLine, int nShowCmd)
{
    LoadLibrary(TEXT("riched20.dll"));
    HWND hwndRTF = CreateWindow(RICHEDIT_CLASS, NULL,
                               ES_MULTILINE | WS_OVERLAPPEDWINDOW,
                               CW_USEDEFAULT, CW_USEDEFAULT,
                               CW_USEDEFAULT, CW_USEDEFAULT,
                               NULL, 0, 0, 0);

    if (hwndRTF) {
        SendMessage(hwndRTF, EM_EXLIMITTEXT, 0, -1);
        if (FillRichEditFromFile(hwndRTF, lpCmdLine)) {
            PRINTDLG pd = { sizeof(pd) };
            pd.Flags = PD_RETURNDC | PD_RETURNDEFAULT;
            if (PrintDlg(&pd)) {
                DOCINFO di = { sizeof(di) };
                di.lpszDocName = TEXT("Sample Printout");
                if (StartDoc(pd.hDC, &di) > 0) {
                    if (PrintRTF(hwndRTF, pd.hDC)) {
                        EndDoc(pd.hDC);
                    } else {
                        AbortDoc(pd.hDC);
                    }
                }
                GlobalFree(pd.hDevMode);
                GlobalFree(pd.hDevNames);
                DeleteDC(pd.hDC);
            }
        }
        DestroyWindow(hwndRTF);
    }
    return 0;
}

```

There's not really much going on here; it's all just glue and necessary typing.

We create a rich edit control and fill it with the file passed on the command line. We then ask the `PrintDlg` function to give us a DC to the user's default printer. We give the document a title, start the document, print the rich text into the document, and then end the document (or abort it if something went wrong during printing). A little cleaning up, and we're all done. A tiny program to print an arbitrary RTF document with no fanfare whatsoever.

See? It's not so hard. Once you find `EM_FORMATRANGE` the rest is just doing the obvious.

Raymond Chen

Follow



