

Determining whether your window is covered

 devblogs.microsoft.com/oldnewthing/20030902-00

September 2, 2003



Raymond Chen

The method described in the previous coding blog entry works great if you are using the window visibility state to control painting, since you're using the paint system itself to do the heavy lifting for you.

To obtain this information outside of the paint loop, use `GetDC` and `GetClipBox`. The HDC that comes out of `GetDC` is clipped to the visible region, and then you can use `GetClipBox` to extract information out of it.

Start with our scratch program and add these lines:

```

void CALLBACK
PollTimer(HWND hwnd, UINT uMsg, UINT_PTR idTimer, DWORD dwTime)
{
    HDC hdc = GetDC(hwnd);
    if (hdc) {
        RECT rcClip, rcClient;
        LPCTSTR pszMsg;
        switch (GetClipBox(hdc, &rcClip)) {
            case NULLREGION:
                pszMsg = TEXT("completely covered"); break;
            case SIMPLEREGION:
                GetClientRect(hwnd, &rcClient);
                if (EqualRect(&rcClient, &rcClip)) {
                    pszMsg = TEXT("completely uncovered");
                } else {
                    pszMsg = TEXT("partially covered");
                }
                break;
            case COMPLEXREGION:
                pszMsg = TEXT("partially covered"); break;
            default:
                pszMsg = TEXT("Error"); break;
        }
        // If we wanted, we could also use RectVisible
        // or PtVisible - or go totally overboard by
        // using GetClipRgn
        ReleaseDC(hwnd, hdc);
        SetWindowText(hwnd, pszMsg);
    }
}

BOOL
OnCreate(HWND hwnd, LPCREATESTRUCT lpcs)
{
    SetTimer(hwnd, 1, 1000, PollTimer);
    return TRUE;
}

```

Once a second, the window title will update with the current visibility of the client rectangle.

Polling is more expensive than letting the paint system do the work for you, so do try to use the painting method first.

(Side note: The reason why part 9 of the scrollbar series is so slow to come out finally struck me as I tried to write it: It's too big. I've split it into parts 9 through 12, with an optional part 13; that may make the little pieces more manageable. Part 9 is written, but I want to hold off posting it until I've got at least through part 12, because something from the later parts may force me to rewrite part 9. A somewhat self-absorbed and rather boring insight into the writing process.)

Raymond Chen

Follow

