

How do I switch a window between normal and fullscreen?

devblogs.microsoft.com/oldnewthing/20100412-00

April 12, 2010



Raymond Chen

Frederic Delhoume wants to know if there is [a simple example of code that switches an application from windowed to fullscreen](#). He then included a code fragment that did some crazy things with parent windows and hiding and showing.

You're making it way, way harder than it needs to be. Let's start with our [scratch program](#) and make these changes:

```
WINDOWPLACEMENT g_wpPrev = { sizeof(g_wpPrev) };
void OnLButtonUp(HWND hwnd, int x, int y, UINT keyFlags)
{
    DWORD dwStyle = GetWindowLong(hwnd, GWL_STYLE);
    if (dwStyle & WS_OVERLAPPEDWINDOW) {
        MONITORINFO mi = { sizeof(mi) };
        if (GetWindowPlacement(hwnd, &g_wpPrev) &&
            GetMonitorInfo(MonitorFromWindow(hwnd,
                MONITOR_DEFAULTTOPRIMARY), &mi)) {
            SetWindowLong(hwnd, GWL_STYLE,
                dwStyle & ~WS_OVERLAPPEDWINDOW);
            SetWindowPos(hwnd, HWND_TOP,
                mi.rcMonitor.left, mi.rcMonitor.top,
                mi.rcMonitor.right - mi.rcMonitor.left,
                mi.rcMonitor.bottom - mi.rcMonitor.top,
                SWP_NOOWNERZORDER | SWP_FRAMECHANGED);
        }
    } else {
        SetWindowLong(hwnd, GWL_STYLE,
            dwStyle | WS_OVERLAPPEDWINDOW);
        SetWindowPlacement(hwnd, &g_wpPrev);
        SetWindowPos(hwnd, NULL, 0, 0, 0, 0,
            SWP_NOMOVE | SWP_NOSIZE | SWP_NOZORDER |
            SWP_NOOWNERZORDER | SWP_FRAMECHANGED);
    }
}
// add to WndProc
HANDLE_MSG(hwnd, WM_LBUTTONDOWN, OnLButtonUp);
```

To avoid getting into the intricacies of hotkeys and accelerators, I opted to toggle to fullscreen on a click. When the button goes up, we check whether we are currently in normal mode or fullscreen mode by sniffing at our window styles. If we are in normal mode, we save the current window placement and get the dimensions of the current monitor. And then the magic happens: We remove the caption and other decorations from our window style and reposition the window so it covers the entire monitor. An important flag to pass here is `SWP_FRAMECHANGED`, which tells the window manager to recalculate the window decorations (which we need it to do because we just changed them).

When returning from fullscreen, we just undo what we had done when we went fullscreen: We restore the caption and other decorations to our window and restore the original window state.

And that's all there is to it. You don't have to do anything special to get the taskbar to "get out of the way"; the taskbar recognizes when applications have gone fullscreen and automatically gets out of the way.

Raymond Chen

Follow

