
Subject: Re: FullScreen: why does not work?
Posted by [alendar](#) on Sun, 05 Dec 2010 03:56:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

I solved this with Win32 APIs. First I tried to use the U++ wrappers but I could not get them to work.

Here's the code:

```
bool isFullScreenModeActive;

virtual bool Key(dword key, int count) {
    static dword style;
    static Rect normalwindowrect;

    switch (key) {
    case K_F11:
        if (!isFullScreenModeActive) {
            // Get the current Top window's style bits
            style = GetStyle();
            // Strips the caption bit, which is the title bar
            style = (style & ~WS_CAPTION);
            // Save the "overlapped" or normal window shape
            normalwindowrect = GetRect();
            SetWindowLong(GetHWND(), GWL_STYLE, style);

            // Get the full size of the screen
            long cx = GetSystemMetrics(SM_CXSCREEN);
            long cy = GetSystemMetrics(SM_CYSCREEN);
            // Expand the window to full size
            SetWindowPos(GetHWND(),HWND_TOP,0,0,cx,cy,SWP_SHOWWINDOW);
            //or use Maximize(false) if you like a
            //delayed resizing, even with zoom effects turned off
        } else {
            // Set the caption bit back so we can get our title bar back
            style |= WS_CAPTION;
            // Pass to windows (Style() method gets confused on SyncCaption0)
            SetWindowLong(GetHWND(), GWL_STYLE, style);
            // Return size to normal; key flag is SWP_FRAMECHANGED
            SetWindowPos(GetHWND(),HWND_TOP,
                normalwindowrect.left,
                normalwindowrect.top,
                normalwindowrect.Width(),
                normalwindowrect.Height(), SWP_SHOWWINDOW|SWP_FRAMECHANGED);
        }
        // Track logically in our application
        isFullScreenModeActive = !isFullScreenModeActive;
        break;
    }
```

```
}  
return false;  
}
```

This was on MS Windows XP Professional V 2002 SP 3

U++ 2791

This is a win32 only fix. This works on a TopWindow while your in it, so you don't have to use the constructor method.

I might try getting the virtual window size so I can zoom across multiple screens.

The trick to getting the title bar back after your zoom is the SWP_FRAMECHANGED flag.

Without this Windoze doesn't know to rebuild the window frame after you added the WS_CAPTION bit back in.

You can use the TopWindow.Maximize() function, but for some reason it still tries to do effects when you pass a false bit, so I used the SetWindowPos instead for the zoom.

I haven't tried this with any windows besides a TopWindow.

This covers the taskbar on the bottom on my computer. Sometimes there is a delay in minimizing the taskbar.

Edit from 5 min later:

To go across all screens just change:

```
cx = GetSystemMetrics(SM_CXVIRTUALSCREEN);  
long cy = GetSystemMetrics(SM_CYSCREEN);  
cy = GetSystemMetrics(SM_CYVIRTUALSCREEN);
```