How do I cover the taskbar with a fullscreen window?

devblogs.microsoft.com/oldnewthing/20050505-04

May 5, 2005

Raymond Chen

For some reason, people think too hard. If you want to create a fullscreen window that covers the taskbar, just create a fullscreen window and the taskbar will automatically get out of the way. Don't go around hunting for the taskbar and poking it; let it do its thing.

As always, start with the scratch program and add the following:

```
HWND CreateFullscreenWindow(HWND hwnd)
{
 HMONITOR hmon = MonitorFromWindow(hwnd,
                                   MONITOR_DEFAULTTONEAREST);
 MONITORINFO mi = { sizeof(mi) };
 if (!GetMonitorInfo(hmon, &mi)) return NULL;
 return CreateWindow(TEXT("static"),
       TEXT("something interesting might go here"),
       WS_POPUP | WS_VISIBLE,
       mi.rcMonitor.left,
       mi.rcMonitor.top,
       mi.rcMonitor.right - mi.rcMonitor.left,
       mi.rcMonitor.bottom - mi.rcMonitor.top,
       hwnd, NULL, g_hinst, 0);
}
void OnChar(HWND hwnd, TCHAR ch, int cRepeat)
{
 if (ch == TEXT(' ')) {
  CreateFullscreenWindow(hwnd);
 }
}
    HANDLE_MSG(hwnd, WM_CHAR, OnChar);
```

Note that this sample program doesn't worry about destroying that fullscreen window or preventing the user from creating more than one. It's just a sample. The point is seeing how the `CreateFullScreenWindow` function is written.

We use the `MonitorFromWindow` function to figure out which monitor we should go fullscreen to. Note that in a multiple monitor system, this might not be the same monitor that the taskbar is on. Fortunately, we don't have to worry about that; the taskbar figures it out.

I've seen people hunt for the taskbar window and then do a `ShowWindow(hwndTaskbar, SW_HIDE)` on it. This is nuts for many reasons.

First is a mental exercise you should always use when evaluating tricks like this: "What if two programs tried this trick?" Now you have two programs both of which think they are in charge of hiding and showing the taskbar, neither of which is coordinating with the other. The result is a mess. One program hides the taskbar, then the other does, then the first decides it's finished so it unhides the taskbar, but the second program wasn't finished yet and gets a visible taskbar when it thought it should be hidden. Things only go downhill from there.

Second, what if your program crashes before it gets a chance to unhide the taskbar? The taskbar is now permanently hidden and the user has to log off and back on to get their taskbar back. That's not very nice.

Third, what if there is no taskbar at all? It is common in Terminal Server scenarios to run programs by themselves without Explorer [archived]. In this configuration, there is no Explorer, no taskbar. Or maybe you're running on a future version of Windows that doesn't have a taskbar, it having been replaced by some other mechanism. What will your program do now?

Don't do any of this messing with the taskbar. Just create your fullscreen window and let the taskbar do its thing automatically.

Raymond Chen

Follow