

# How can I make the touch keyboard appear automatically when focus enters an edit control in my program?

 [devblogs.microsoft.com/oldnewthing/20150601-00](http://devblogs.microsoft.com/oldnewthing/20150601-00)

June 1, 2015



Raymond Chen

By default, the Windows 8 touch keyboard does not appear automatically when focus is placed on an edit control in a desktop program. To change the behavior for your program, just use this one weird trick:

```
HRESULT EnableTouchKeyboardFocusTracking()
{
    ComPtr<IInputPanelConfiguration> configuration;
    HRESULT hr =
        CoCreateInstance(__uuidof(InputPanelConfiguration), nullptr,
            CLSCTX_INPROC_SERVER, IID_PPV_ARGS(&configuration));
    if (SUCCEEDED(hr)) {
        hr = configuration->EnableFocusTracking();
    }
    return hr;
}
```

You create an instance of the `InputPanelConfiguration` object and ask it to enable focus tracking. This is a per-process setting, and once set, it cannot be unset.

Let's use this function in a Little Program so you can play with it. Most of the work in setting up the program is creating two controls: an edit control and a button. If I had just one control, then you wouldn't be able to see how the keyboard automatically appears and disappears when focus moves between an edit control and some other type of control.

Remember that Little Programs do little to no error checking. Start with [the scratch](#) program and make these changes:

```

#define STRICT
#include ...
#include <shobjidl.h>
#include <inputpanelconfiguration.h>
#include <wrl\client.h>
#include <wrl\event.h>

using namespace Microsoft::WRL;

HINSTANCE g_hinst; /* This application's HINSTANCE */
HWND g_hwndChild; /* Optional child window */
HWND g_hwndButton;
HWND g_hwndLastFocus;

void
DoLayout(HWND hwnd, int cx, int cy)
{
    if (g_hwndChild) {
        MoveWindow(g_hwndChild, 0, 0, cx - 100, cy, TRUE);
    }
    if (g_hwndButton) {
        MoveWindow(g_hwndButton, cx - 100, 0, 100, 50, TRUE);
    }
}

void
OnSize(HWND hwnd, UINT state, int cx, int cy)
{
    DoLayout(hwnd, cx, cy);
}

BOOL
OnCreate(HWND hwnd, LPCREATESTRUCT lpcs)
{
    g_hwndChild = CreateWindow(TEXT("edit"), nullptr,
        WS_CHILD | WS_VISIBLE | WS_BORDER | ES_MULTILINE,
        0, 0, 100, 100, hwnd, nullptr, g_hinst, 0);
    g_hwndButton = CreateWindow(TEXT("button"), TEXT("Send"),
        WS_CHILD | WS_VISIBLE | BS_PUSHBUTTON,
        0, 0, 100, 100, hwnd, nullptr, g_hinst, 0);

    EnableTouchKeyboardFocusTracking();
    return TRUE;
}

// OnActivate incorporated by reference.

HANDLE_MSG(hwnd, WM_ACTIVATE, OnActivate);

BOOL
InitApp(void)
{

```

```
...  
wc.hbrBackground = (HBRUSH)(COLOR_APPWORKSPACE + 1);  
...  
}
```

We position the edit control on the left hand side of the window and put the button in the upper right corner. We enable focus tracking on the touch keyboard, and just to make it easier to see where the edit control is, we give the frame with the app-workspace color.

Although we summon the touch keyboard when focus enters the edit control, we do nothing to prevent the keyboard from covering what the user is typing. This is one of the reasons that the touch keyboard does not appear automatically when focus is placed in an edit control of a desktop program. It would end up covering the edit control the user is trying to type into!

We'll work on fixing this problem next week.

Raymond Chen

**Follow**

