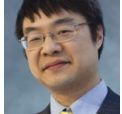


Why doesn't the "Automatically move pointer to the default button in a dialog box" work for nonstandard dialog boxes, and how do I add it to my own nonstandard dialog boxes?

 devblogs.microsoft.com/oldnewthing/20130826-00

August 26, 2013



Raymond Chen

The Mouse control panel has a setting called *Automatically move pointer to the default button in a dialog box*, known informally as *Snap to default button* or simply *Snap To*. You may have discovered that it doesn't work for all dialog boxes. Why not?

The *Snap To* feature is implemented by the dialog manager. When the window is shown and the setting is enabled, it will center the pointer on the default button. If your application does not use the dialog manager but instead creates its own custom dialog-like windows, then naturally the code in the standard dialog manager will not run.

If you want your nonstandard dialog box to support the *Snap To* feature, you get to implement it yourself.

Here's a Little Program that creates a window with a default pushbutton inside it, and which centers the mouse on the button when the window is shown. Start with our [scratch program](#) and make these changes:

```
POINT GetRectCenter(LPCRECT prc)
{
    POINT pt = {
        prc->left + (prc->right - prc->left) / 2,
        prc->top + (prc->bottom - prc->top) / 2
    };
    return pt;
}
```

The `GetRectCenter` helper function calculates center of a rectangle.

```

BOOL OnCreate(HWND hwnd, LPCREATESTRUCT lpcs)
{
    CreateWindow(TEXT("button"),
                TEXT("Button 1"),
                WS_CHILD | WS_VISIBLE | BS_DEFPUSHBUTTON,
                0, 0, 200, 50,
                hwnd,
                (HMENU)1,
                g_hinst,
                0);
    return TRUE;
}

```

When our main window is created, we put a default button inside it.

```

#define WM_CHECKSNAPDEFBUTTON WM_APP
void OnShowWindow(HWND hwnd, BOOL fShow, UINT status)
{
    if (fShow && status == 0)
    {
        PostMessage(hwnd, WM_CHECKSNAPDEFBUTTON, 0, 0);
    }
}
void OnCheckSnapDefButton(HWND hwnd)
{
    BOOL fSnapToDefButton;
    if (SystemParametersInfo(SPI_GETSNAPTODEFBUTTON, 0,
                            &fSnapToDefButton, 0) &&
        fSnapToDefButton &&
        hwnd == GetForegroundWindow())
    {
        RECT rcButton;
        GetWindowRect(GetDlgItem(hwnd, 1), &rcButton);
        POINT ptCenter = GetRectCenter(&rcButton);
        SetCursorPos(ptCenter.x, ptCenter.y);
    }
}

```

When the window is shown, we post a message to check the *Snap To* setting after things have settled down. Once things settle down, we check the *Snap To* setting, and if it's enabled, and if our window is still the foreground window, then we center the cursor on our button.

It's important to check that our window is still the foreground window, because it would be rude to move the cursor to our button even if we opened in the background.

That's why we need to post a message to perform the check later. The `WM_SHOWWINDOW` message is sent early in the `ShowWindow` calculations, before the activation actually changes. If we performed the check then, the answer would always be, "No, you're not the foreground window," and we would always back off.

Naturally, we need to hook up our new messages.

```
HANDLE_MSG(hwnd, WM_SHOWWINDOW, OnShowWindow);
case WM_CHECKSNAPDEFBUTTON: OnCheckSnapDefButton(hwnd); break;
```

And there you have it, a program that honors the *Automatically move pointer to the default button in a dialog box* setting in its custom nonstandard dialog.

Exercise: What assumptions are made about the rectangle by the `GetRectCenter` function? How do they differ from the assumptions made by this alternate version:

```
POINT GetRectCenter(LPCRECT prc)
{
    POINT pt = {
        (prc->left + prc->right) / 2,
        (prc->top + prc->bottom) / 2
    };
    return pt;
}
```

Raymond Chen

Follow

