

# Is this a really bug with CreateWindowEx or am I just confused?

---

 [devblogs.microsoft.com/oldnewthing/20110912-00](http://devblogs.microsoft.com/oldnewthing/20110912-00)

September 12, 2011



Raymond Chen

Somebody sent me email pointing out strange behavior in the `MessageBox` function if you fail a window creation by returning `-1` from the `WM_CREATE` message. On the other hand, returning `FALSE` from `WM_NCCREATE` seems to work just fine. “So why the difference with `WM_CREATE` ?”

```

#include <windows.h>
LRESULT CALLBACK
WndProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    switch(uMsg)
    {
        case WM_CREATE:
            return -1;
        case WM_DESTROY:
            PostQuitMessage(0);
            return 0;
    }
    return DefWindowProc(hWnd, uMsg, wParam, lParam);
}
int APIENTRY WinMain(HINSTANCE hInst, HINSTANCE hPrev,
    LPSTR lpCmdLine, int nShowCmd)
{
    MSG msg;
    HWND hWnd;
    WNDCLASS wc = { 0 };
    wc.lpfnWndProc = WndProc;
    wc.hInstance = hInst;
    wc.hIcon = LoadIcon(NULL, IDI_APPLICATION);
    wc.hCursor = LoadCursor(NULL, IDC_ARROW);
    wc.hbrBackground = (HBRUSH)(COLOR_WINDOW + 1);
    wc.lpszClassName = "TestApp";
    if(!RegisterClass(&wc)){
        MessageBox(NULL, "Error creating class",
            "Test App", MB_ICONERROR);
        return 1;
    }
    hWnd = CreateWindow(wc.lpszClassName, "Test App",
        WS_OVERLAPPEDWINDOW, CW_USEDEFAULT, CW_USEDEFAULT,
        CW_USEDEFAULT, CW_USEDEFAULT, NULL, 0, hInst, NULL);
    if(!hWnd){
        MessageBox(NULL, "Error creating window",
            "Test App", MB_ICONERROR);
        return 1;
    }
    ShowWindow(hWnd, nShowCmd);
    UpdateWindow(hWnd);
    while(GetMessage(&msg, NULL, 0, 0))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return (int)msg.wParam;
}

```

You already know enough to solve this puzzle. You just need to connect the dots.

(In fact, the person who sent me this topic did so a year after I already answered it. But I'm repeating it here because the original answer was accidentally destroyed.)

Raymond Chen

**Follow**

