

Re: window from win32 dll

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2008-09/msg00019.html>

- *From:* Joseph M. Newcomer <newcomer@xxxxxxxxxxxxx>
 - *Date:* Mon, 01 Sep 2008 11:06:17 -0400
-

See below...

On Mon, 1 Sep 2008 04:45:21 -0700 (PDT), keralafood@xxxxxxxxxxx wrote:

```
I really fed up with this problem,i have to create one window from
win32 DLL,
BOOL Derived::InitWindow(int left,int top,int width,int
height,HINSTANCE h)
{
```

Snip

```
WNDCLASSEX wc = { sizeof(WNDCLASSEX), CS_CLASSDC, MsgProc, 0L,
0L,
h, NULL, NULL, NULL, NULL,
L"Renderer", NULL };
RegisterClassEx( &wc );
```

Move these two lines to your DllMain/DLL_PROCESS_ATTACH handler. I'm not sure why you would write "0L" when "0" is sufficient. Also, what is h? It should be the instance handle of the DLL, according to the documentation. Why aren't you passing in a const CRect& instead of explicit width and height?

I notice that you seem to have failed to see if this actually worked; I did not see anything of the form

```
ATOM at = ::RegisterClassEx(&wc);
if(at == NULL)
return FALSE;
so how do you know this worked?
```

But after moving it to DllMain/DLL_PROCESS_ATTACH, you will then add

```
case DLL_PROCESS_DETACH:
UnregisterClassEx(L"Renderer");
return TRUE;
```

Re: window from win32 dll

Also, is there some reason you are using CS_CLASSDC, which is a very, very strange technique, and probably extremely dangerous to use. Is there some reason you want every instance of the window to share the one-and-only DC that is a class DC? (This style probably hasn't made sense since Win32 came out, since its purpose was to reduce GDI memory usage on an 8088 running Windows 1.0)

Note that returning FALSE from DllMain is generally poor practice in production code, but it takes a 10-minute lecture to explain why, and what you have to do to get around it, which I don't want to go into right now.

```
// Create the application's window
_hRenderWnd = CreateWindow(L"Renderer",0,
WS_POPUPWINDOW, left, top, width, height,
NULL, NULL, h,
NULL);
return TRUE;
}
```

```
msgProc declared declared in Derived class.static LRESULT WINAPI
MsgProc( HWND hWnd, UINT msg, WPARAM wParam, LPARAM lParam );
```

A strange thing to want to do; why are you declaring a window handler in a derived class, when you could just as easily use MFC in a non-extension DLL? Note that using this class in the .exe file can lead to potential confusion. The handler for your messages has to be in the DLL (or why bother using a DLL). What kind of class are you using? CWnd? If CWnd, why are you mixing raw Win32 API with MFC?

I note that you have not actually said anything about what is going wrong, or if anything failed, what the GetLastError values are. For example, you are presuming RegisterClassEx succeeds without actually testing that it does, or that the CreateWindow works, without actually testing that it does.

Perhaps you are seeing no window? Is that what is happening? If you don't say WS_VISIBLE as a window style, it happily creates an invisible window. You have neglected to explain what failures you are seeing, which is why testing the RegisterClassEx and CreateWindow calls is sort of critical to the analysis...all you've said is "my program doesn't work" followed by some highly questionable code, but given no basis for the analysis.

```
.HINSTANCE storing in dllmain HMODULE
```

```
CreateWindow always fail..
```

```
i tried this way also
```

```
WNDCLASSEX wc = { sizeof(WNDCLASSEX), CS_CLASSDC|CS_HREDRAW |
```

Re: window from win32 dll

Re: window from win32 dll

Please try to put spaces around binary operators

```
CS_VREDRAW | CS_OWNDC | CS_GLOBALCLASS, MsgProc, 0L, 0L,  
GetModuleHandle(NULL), NULL, NULL, NULL, NULL,  
szWindowClass, NULL };  
RegisterClassEx( &wc );
```

GetModuleHandle(NULL) returns the module handle of the executable, but your handler function is in the DLL, so why do you go out of your way to make the same error as in the first case? You want to save the HMODULE/HINSTANCE parameter of DllMain and use that here, as the documentation carefully states.

```
// Create the application's window  
hwnd = CreateWindow(szWindowClass,L"Title",  
WS_POPUPWINDOW| CS_GLOBALCLASS, left, top, width,  
height,  
GetDesktopWindow(), NULL,GetModuleHandle(NULL),  
NULL);
```

Is szWindowClass set to the same value you used when you did the RegisterClassEx. I see no evidence here that it is.

It is always a mistake to create a window as a child of the desktop window; take that as your guideline and the few exceptions will be obvious.

Have you tried

```
hwnd = ::CreateWindow(szWindowClass, // window class  
NULL, // caption  
WS_POPUPWINDOW | WS_VISIBLE,  
left, top, width, height,  
parent,  
NULL,  
dllsinstance,  
NULL);  
if(hwnd == NULL)  
return FALSE;
```

There is no such thing as a "static dll". A dll is either implicitly linked dynamically or explicitly linked dynamically, and from your viewpoint there is no difference between these two styles.

```
return TRUE;  
but all are avail..  
this dll is a static dll(lib and header file)  
any rule for creating window inside dll?
```

Re: window from win32 dll

Re: window from win32 dll

i must create another thread ?should i call in dllmain?

You can't call it from DllMain. RTFM. Creating from another thread is somewhere between silly and dangerous; if you can't create it using the code you have, the code you have is wrong. Note that you also cannot create a window in a function *called* from DllMain.

please help me ..

NB: second post (afterwin32 UI)

Joseph M. Newcomer [MVP]

email: newcomer@xxxxxxxxxxxxx

Web: <http://www.flounder.com>

MVP Tips: http://www.flounder.com/mvp_tips.htm

.