
Subject: Re: CALL_EXTERNAL

Posted by [menakkis](#) on Mon, 22 Feb 1999 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

> Hello!
> Help me please!
> I called function CALL_EXTERNAL and recieved this message:
> ERROR_MOD_NOT_FOUND
> What is it mean and what's my mistake?
> code is:
> CALL_EXTERNAL('DllProject.dll','bit_operation',buffer[pos+6])
> output:
> % CALL_EXTERNAL: Error loading sharable executable.
> Symbol: bit_operation, File = ./DllProject.dll
> ERROR_MOD_NOT_FOUND
> in dll exist export function with name "_bit_operation"(checked in QuikView)

Sounds like you'd be quite pleased to get this working :-). Sorry for the delay, but there's been a week-end. (In other words, my apologies to the NG for my previous flippant post.)

I assume that you're using C. In case you haven't got it all working yet, the short answer is: Declare your function as WINAPI (which translates to `__stdcall`), and export it with a .DEF file, like in the example that comes with IDL. (You have to do a little digging to discover that `__stdcall` is used in the examples.) If you're using visual C then just plonk the .DEF file amongst your c and h files via "Add to project" -> "Files...".

Now there are other ways to get CALL_EXTERNAL to work, especially since IDL started supporting the "cdecl" calling convention as an option. Myself, I haven't experimented in this area for a while, as I found the .DEF file method works in all IDL versions that are win32-based. But here are some suggestions... If you don't mind making your IDL code more Windows-specific, you could alternatively try sticking an underscore on the front of the function's name in your IDL code. If the function has been exported with only this name decoration AND (somehow) built to use the "stdcall" calling convention then it should work. If the "cdecl" calling convention, then you'll also have to use the /CDECL switch in CALL_EXTERNAL(). If you have exported your function by declaring it as `__declspec(dllexport)` (which has been the Microsoft- recommended way for quite some time now) then I think it'll use cdecl, but I'm not really sure. (Some dark corners of the MS documentation allude to the calling convention being optional here, like it'll go with the default if you don't add "`__stdcall`". It's rather obscure :-() I also don't know whether IDL will do you the favour of trying a leading underscore with /CDECL if the plain name doesn't work - it should do as a leading underscore is cdecl's default way of decorating names, apparently. Give it a try.

Peter Mason

-----== Posted via Deja News, The Discussion Network ==-----
<http://www.dejanews.com/> Search, Read, Discuss, or Start Your Own
