
Subject: calling function in DLL compiled with C++
Posted by [Dominik\[1\]](#) on Mon, 07 Jan 2002 13:11:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi NG,

I tried to write a DLL using C++. But I have a big problem, calling the function in the DLL out of IDL, when the DLL is compiled using a C++-compiler. Using a C-compiler, everything is fine, there is no problem, calling the function.

Does somebody know why? Is there a difference in writing a DLL for IDL in C and in C++?

Or does somebody have an DLL-example in C++?

Thanks for your help
Dom

Subject: Re: calling function in DLL compiled with C++
Posted by [Gert Van de Wouwer](#) on Tue, 08 Jan 2002 10:46:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I struggled with this problem too. The problem is probably to be solved by changing the build parameters of your dll.
I use IDL on Windows 2000 and Visual Studio 6.0. I use an exports.def file to export the dll symbols (rather than declspec-things)
see example below - hope this helps

Gert

exports.def:

```
-----  
LIBRARY LocMinDll  
DESCRIPTION 'ILD dll LocMin (c)GVDW'
```

```
EXPORTS  
  GetVersion  
  LocMinDll
```

```
-----  
the code goes in a cpp file
```

```
-----  
IDL_LONG IDL_STDCALL LocMinDll( int argc, void* argv[])
```

```
{  
<your super code here>  
}
```

"Dominik Paul" <dpaul@ukl.uni-freiburg.de> wrote in message
news:a1c6k0\$16f\$1@n.ruf.uni-freiburg.de...

> Hi NG,
>
> I tried to write a DLL using C++. But I have a big problem, calling the
> function in the DLL out of IDL, when the DLL is compiled using a
> C++-compiler. Using a C-compiler, everything is fine, there is no problem,
> calling the function.
>
> Does somebody know why? Is there a difference in writing a DLL for IDL in
C
> and in C++?
> Or does somebody have an DLL-example in C++?
>
> Thanks for your help
> Dom
>
>

Subject: Re: calling function in DLL compiled with C++
Posted by [Richard Younger](#) on Thu, 10 Jan 2002 22:11:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dominik Paul wrote:

>
> Hi NG,
>
> I tried to write a DLL using C++. But I have a big problem,
> calling the function in the DLL out of IDL, when the DLL is
> compiled using a C++-compiler.
[...]
>
> Dom

Hi, Dom.

Are you still having trouble? I've got a few examples lying around for
VC++, but I don't know what compiler you're using. I certainly second
Gert's suggestion about using .def files if you're in VC++.

If things still aren't working, you can use the extern "C" construct to force your C++ compiler to use C names and calling conventions in most compilers.

Just add something like

```
#ifdef __cplusplus
extern "C" {
#endif

/* C callable prototypes here */

#ifdef __cplusplus
}
#endif
```

to your header file around IDL_load() or the externally visible functions. More info at <http://www.parashift.com/c++-faq-lite/mixing-c-and-cpp.html>.

Best,
Rich

--
Richard Younger

Subject: Re: calling function in DLL compiled with C++
Posted by [Dominik\[1\]](#) on Mon, 14 Jan 2002 08:01:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Richard,

I already tried the extern C construct. But then I got some problems, using other C++ routines in the DLL.
So, this method didn't work for me.

But Greg's suggestion was ok. The DLL is running now, thanks
Dom

"Richard Younger" <younger@ll.mit.edu> schrieb im Newsbeitrag
news:3C3E119B.8A355D5B@ll.mit.edu...

> Dominik Paul wrote:
>>
>> Hi NG,
>>
>> I tried to write a DLL using C++. But I have a big problem,
>> calling the function in the DLL out of IDL, when the DLL is
>> compiled using a C++-compiler.
> [...]
>>
>> Dom
>
> Hi, Dom.
>
> Are you still having trouble? I've got a few examples lying around for
> VC++, but I don't know what compiler you're using. I certainly second
> Gert's suggestion about using .def files if you're in VC++.
>
> If things still aren't working, you can use the extern "C" construct to
> force your C++ compiler to use C names and calling conventions in most
> compilers.
>
> Just add something like
>
>
> #ifdef __cplusplus
> extern "C" {
> #endif
>
> /* C callable prototypes here */
>
> #ifdef __cplusplus
> }
> #endif
>
>
> to your header file around IDL_load() or the externally visible
> functions. More info at
> <<http://www.parashift.com/c++-faq-lite/mixing-c-and-cpp.html>>.
>
>
> Best,
> Rich
>
> --
> Richard Younger

Subject: Re: calling function in DLL compiled with C++
Posted by [Richard Younger](#) on Thu, 24 Jan 2002 20:44:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dominik Paul wrote:

>
> Hi Richard,
>
> I allready tried the extern C construct. But then I got some problems,
> using other C++ routines in the DLL.
> So, this method didnt work for me.
>
> But Gregs suggestion was ok. The DLL is running now, thanks
> Dom

Hi Dom,

Glad you got everything working.

As an addendum, I was looking something up in my Windows programming reference, and I stumbled upon something of interest. It seems that even in straight C, the MSVC compiler mangles names (prepending an underscore and tacking a @[# of bytes in parameters] onto the end) if you're using the `__stdcall` calling convention, even if you extern "C" your function calls. And yes, I agree that it's annoying.

The upshot is that using `__declspec(dllexport)` and `__stdcall`, you'd have to call something like `_MyGenData@12` instead of `MyGenData`. So, you can either use a .def file like you are or include something like

```
#pragma comment(linker, "/export:MyGenData=_MyGenData@12")
```

to alias `MyGenData` to `_MyGenData@12` in the DLL exports.

It seems like more trouble and work to do it this way than a .def file, but I guess for anyone who wants to support multiple architecture source, they can place it behind an `#ifdef WIN32` or whatever, instead of having to deal with the extra file and compiler options.

Just thought somebody somewhere might find some passing interest.

Best,
Rich

--

Richard Younger
Phone: (781)981-4464
Fax: (781)981-0122

MIT Lincoln Laboratory
244 Wood St.
Lexington, MA 02421

Subject: Re: calling function in DLL compiled with C++
Posted by [wbiagiot](#) on Wed, 06 Feb 2002 17:00:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Richard Younger <younger@ll.mit.edu> wrote in message
news:<3C50723F.E0D299F9@ll.mit.edu>...

> Dominik Paul wrote:

>>

>> Hi Richard,

>>

>> I allready tried the extern C construct. But then I got some problems,

>> using other C++ routines in the DLL.

>> So, this method didnt work for me.

>>

>> But Gregs suggestion was ok. The DLL is running now, thanks

>> Dom

>

I just stumbled across this thread and wanted to remind everyone about
the TERRIFIC free DLM that Randall Frank has on Ronn Kling's website.
Here's a snipit about the relevant section:

* A generic interface for calling Windows DLL functions directly

This interface allows an IDL program to directly access the
exported functions in any Windows DLL. This interface can
replace the CALL_EXTERNAL wrappers for many functions under
Windows. The interface consists of two functions. One to
define the external function to IDL and one to query the
external function table.

I have used this interface with 3rd party DLLs (i.e. no source code)
with great success (National Instrument's VISA library). I consider
this DLM an essential part of my IDL code (gush gush). Check it out.

-Bill
