
Subject: Using MSVC++ 4.2 to create DLL for CALL_EXTERNAL

Posted by [Richard Tyc](#) on Wed, 12 Jan 2000 08:00:00 GMT

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

I am new to Windows development and am having some problems creating a very simple DLL within Microsoft Developer Studio to be called my IDL using CALL_EXTERNAL. I have been discussing the problem with RSI tech support and they supplied me with the following simple program to try within Microsoft Developer Studio (v. 4.2) which I cannot get to compile. They use MSVC++ 6.0 so they are unable to help further.

string_test.c:

```
#include <stdio.h>
#include "e:\rsi\idl52\external\export.h"
#ifdef WIN32
#include <windows.h>
#define IDL_LONG_RETURN __declspec(dllexport) int
#else
#define IDL_LONG_RETURN int
#endif

IDL_LONG_RETURN string_test(int argc, void *argv[])
{
    IDL_STRING idl_str;
    char str_idl[80];

    strcpy(str_idl, "Data Collection\0");
    IDL_StrStore (&idl_str, str_idl);
    printf ("I guess it worked?!?\n");
    return (1L);
}
```

I am building the DLL within MSVC++ as a new project workspace->dynamic-link library. I then add the library idl32.lib with :

Build->Settings-> Link Tab - Category: General added idl32.lib under Object/Library modules

Then I added the path under

Tools->Options-> directories Tab-> Show:Library Files added e:\rsi\idl52

BTW, other libraries also included by default :

kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib
shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib

when I build , I get the following error now:

Linking...

E:\rsi\IDL52\idl32.lib : fatal error LNK1106: invalid file or disk full:
cannot seek to 0x3758b06a
Error executing link.exe.
testDLLrsi.dll - 1 error(s), 0 warning(s)

E: drive has 650Mb space left.???

However, I notice in the examples (.../external/call_external/C) the
makefile_win.mak includes the file : linclude <win32.mak> .. Do I need to
add this to the MSVC++ generated makefile for my project because I notice it
is not in there (the nmake file generated does expressly state DO NOT EDIT
so I was leary of changing it.

Anyone have any hints as to what is going on ??

Thanks in Advance

Rich

Subject: Re: Using MSVC++ 4.2 to create DLL for CALL_EXTERNAL
Posted by [justin_ashmall](#) on Thu, 13 Jan 2000 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

[posted and mailed]
richt@sbrc.umanitoba.ca (Richard Tyc) wrote in
<85i7ag\$hda\$1@canopus.cc.umanitoba.ca>:

> I am new to Windows development and am having some problems creating a
> very simple DLL within Microsoft Developer Studio to be called my IDL
> using CALL_EXTERNAL. I have been discussing the problem with RSI tech
> support and they supplied me with the following simple program to try
> within Microsoft Developer Studio (v. 4.2) which I cannot get to
> compile. They use MSVC++ 6.0 so they are unable to help further.

I'm using 6 now, but I'm pretty sure the following compiled under 5 if not
4.

I'm not sure what your problem might be but you might try my first test
effort, it's very simple and somewhat different to the routine you posted.

> I am building the DLL within MSVC++ as a new project workspace->dynamic
> -link library. I then add the library idl32.lib with :

I didn't add any extra libraries, just included windows.h. I created an
empty workspace and added two files, Test_DLL.cpp (the code) and
Test_DLL.def. The latter tells the compiler which functions you want
exported in the DLL (it also prevents the exported function names being
changed (aka 'decoration')). Make sure the files have been added in to the

project.

Test_DLL.cpp :

```
#include <windows.h>
```

```
// Function prototypes.
```

```
BOOL WINAPI DllMain(HINSTANCE hInst, ULONG ulReason, LPVOID lpReserved);
```

```
long WINAPI TimesTwo(int argc, void* argv[]);
```

```
// Windows DLL entry point.
```

```
BOOL WINAPI DllMain(HINSTANCE hInst, ULONG ulReason, LPVOID lpReserved)
```

```
{  
    return(TRUE);  
}
```

```
// Function to multiply number by two.
```

```
// Number to multiply is passed in the first argument (float)
```

```
// Answer returned in second argument (float)
```

```
long WINAPI TimesTwo(int argc, void* argv[]){
```

```
    //argc holds number of parameters passed
```

```
    //argv is array of pointers to passed parameters, i.e.
```

```
    //argv[0]=pointer to first arg, argv[1] pointer to second, etc
```

```
    //Define two pointers to floats
```

```
    float *multiply_me, *answer;
```

```
    multiply_me=(float *)argv[0];
```

```
    answer=(float *)argv[1];
```

```
    *answer = (*multiply_me) * 2.0;
```

```
    return(0L); //Return zero if all okay
```

```
}
```

Test_DLL.def:

```
LIBRARY Test_DLL
```

```
DESCRIPTION 'DLL for use with IDL'
```

```
EXPORTS TimesTwo
```

The IDL code used to call the DLL:

```
pro test_dll

dll='D:\C++ projects\Test_DLL\Release\Test_DLL.dll'

a=FLOAT(10.21) ;Ensure we have floats to pass to DLL
b=FLOAT(1)

PRINT, a, b ;Before DLL

result = CALL_EXTERNAL (dll, 'TimesTwo', a, b)

PRINT, Result ;Should be zero
PRINT, a, b ;After DLL

end
```

And the output:

```
IDL> test_dll
  10.2100   1.00000
      0
  10.2100  20.4200
```

Note that if you want to export more than one function in the DLL you list them in the .def file as follows:

```
LIBRARY Test_DLL
DESCRIPTION 'DLL for use with IDL'
EXPORTS TimesTwo
        TimesThree
        TimesFour
```

Good Luck!

Justin
