
Subject: Re: Windows IDL DLM with Cygwin
Posted by [Mark Hadfield](#) on Tue, 12 Jul 2005 03:55:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

joeymu@gmail.com wrote:

> Has anyone sucessfully used a DLM created with the Cygwin family of
> compilers? I have sucessfully compiled a DLM and it looks like it
> should work, but when it gets to the point of actually using the DLM,
> IDL will crash!

That's what I would expect. There are problems in using the Cygwin DLL,
or DLLs that link to it, with non-Cygwin applications because of
problems with run-time library initialization. I've done a little
Googling to find a reference for this fact (and I use the term loosely)
but all I've managed to do is confuse myself.

> I've been trying to compile my code with a
> -mno-cygwin, but that is going to be a lot of work! I'm hoping there
> is an easier way. Anyway, I'd love to hear even if this is possible
> (i.e. someone has actually done it) so I don't just spin my wheels.

Hmm. I don't know what is the line of least resistance here. It depends,
partly, of how much work it would be to wean your code off the Cygwin
DLL. Assuming (as you say) that this is going to be a *lot* of work then
perhaps you should try to get the Cygwin run-time initialisation issues
sorted out. You should read this:

http://cygwin.com/faq/faq_3.html#SEC102

Perhaps you should ask about this on the Cygwin mailing list, but read
the relevant sections in the IDL External Development Guide first. (And
be ready for some brusque replies.)

And when you've sorted it all out, perhaps you could report back to the
list?

--

Mark Hadfield "Kei puwaha te tai nei, Hoea tahi tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: Windows IDL DLM with Cygwin
Posted by [Haje Korth](#) on Tue, 12 Jul 2005 12:49:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

Joey,

I tried that years ago, but was never successful. However, in the meantime RSI has put out a TechTip on how to compile DLMs with cygwin. You should dig that up and see if you find what you are doing wrong. I just got tired of poking around to find a fix. You can spend a lot of time on this search. And if it is really true that "time is money", my solution to get Visual Studio is suddenly not that expensive...

Good luck,
Haje

<joeymu@gmail.com> wrote in message
news:1121121297.794813.114820@o13g2000cwo.googlegroups.com.. .
> Has anyone sucessfully used a DLM created with the Cygwin family of
> compilers? I have sucessfully compiled a DLM and it looks like it
> should work, but when it gets to the point of actually using the DLM,
> IDL will crash! I've been trying to compile my code with a
> -mno-cygwin, but that is going to be a lot of work! I'm hoping there
> is an easier way. Anyway, I'd love to hear even if this is possible
> (i.e. someone has actually done it) so I don't just spin my wheels.
>
> Thanks!
>
> Joey
>

Subject: Re: Windows IDL DLM with Cygwin
Posted by [Haje Korth](#) on Tue, 12 Jul 2005 12:51:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Joey, this is the article I was talking about. Haje

Topic:

Some Windows developers choose to use the Cygwin GCC port as their compiler. This TechTip describes how to set up IDL so that CALL_EXTERNAL works with GCC.

Discussion:

Several customers have reported success using CALL_EXTERNAL with Cygwin's GCC Windows port instead of Microsoft Visual Studio.

a.. Create a new user environment variable (Control Panel -> System -> Advanced -> Environment Variables) called IDL_DLM_PATH.
Set this to: C:\RSI\IDL61\bin\bin.x86;c:\cygwin\bin

b.. Set the following parameters on !MAKE_DLL:

1.. !make_dll.LD = 'c:\cygwin\bin\ld.exe -shared -o %L %O %X'

2.. !make_dll.CC = 'c:\cygwin\bin\gcc.exe
%X -fPIC -I"/cygdrive/c/RSI/IDL61/external/include" -c -D_REENTRANT %C -o
%O'

3.. !make_dll.COMPILER_NAME='GCC'

c.. To test, load and compile the STRING_ARRAY.PRO example in the
IDL61\external\call_external\C subdirectory.

IDL> string_array, 'Hello, Cygwin!', /VERBOSE

Calling string_array with:

STRARR_L STRING = 'Hello, Cygwin!'

Hello, Cygwin!

Result from string_array is:

RESULT STRING = 'Hello, Cygwin!'

If the PATH environment does not also include c:\cygwin\bin, users may
receive an error message that cygwin1.dll could not be found.

<joeymu@gmail.com> wrote in message

news:1121121297.794813.114820@o13g2000cwo.googlegroups.com.. .

> Has anyone sucessfully used a DLM created with the Cygwin family of

> compilers? I have sucessfully compiled a DLM and it looks like it

> should work, but when it gets to the point of actually using the DLM,

> IDL will crash! I've been trying to compile my code with a

> -mno-cygwin, but that is going to be a lot of work! I'm hoping there

> is an easier way. Anyway, I'd love to hear even if this is possible

> (i.e. someone has actually done it) so I don't just spin my wheels.

>

> Thanks!

>

> Joey

>

Subject: Re: Windows IDL DLM with Cygwin

Posted by [Karsten Rodenacker](#) on Tue, 12 Jul 2005 14:28:40 GMT

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

Hmm, ja, I am using MinGW compiler under cygwin for creation of dll's (for call external)

It works well, but I don't know the differences between MinGW and Cygwin compiler. You have to look under <http://www.mingw.org/>.

Attached the output of a make (using mingw32-make) where the dll is functional:

```
$ make
```

```
gcc -c -DBUILDING_DLL=1 -I. -g -Wall -o isop2.o isop2.c
```

```
gcc -c -DBUILDING_DLL=1 -I. -g -Wall -o kont2i.o kont2i.c
```

```
dllwrap --export-all --def my_share.def_ALT --output-def my_share.def
```

```
--implib my_share.lib --driver-name gcc -o my_share.dll \
```

```
isop2.o kont2i.o
```

```
$ which gcc
```

```
/cygdrive/d/Program Files/GNU/MinGW/bin/gcc
```

```
$ which dllwrap
```

```
/cygdrive/d/Program Files/GNU/MinGW/bin/dllwrap
```

```
$ which make
```

```
/cygdrive/d/Program Files/GNU/MinGW/bin/make
```

Regards

Karsten

Am Tue, 12 Jul 2005 01:02:09 +0200 schrieb <joeymu@gmail.com>:

> Has anyone successfully used a DLM created with the Cygwin family of
> compilers? I have successfully compiled a DLM and it looks like it
> should work, but when it gets to the point of actually using the DLM,
> IDL will crash! I've been trying to compile my code with a
> -mno-cygwin, but that is going to be a lot of work! I'm hoping there
> is an easier way. Anyway, I'd love to hear even if this is possible
> (i.e. someone has actually done it) so I don't just spin my wheels.

>

> Thanks!

>

> Joey

>

--

Erstellt mit Operas revolutionärem E-Mail-Modul: <http://www.opera.com/m2/>

Subject: Re: Windows IDL DLM with Cygwin
Posted by [joey](#) on Tue, 12 Jul 2005 21:26:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mark Hadfield <m.hadfield@niwa.co.nz> wrote:

> joeymu@gmail.com wrote:

>> Has anyone sucessfully used a DLM created with the Cygwin family of
>> compilers? I have sucessfully compiled a DLM and it looks like it
>> should work, but when it gets to the point of actually using the DLM,
>> IDL will crash!
>
> That's what I would expect.

What I find odd is that IDL just crashes rather than give me a message saying something about a missing symbol or the like. That's really what I was expecting.

>> I've been trying to compile my code with a
>> -mno-cygwin, but that is going to be a lot of work! I'm hoping there
>
> Hmm. I don't know what is the line of least resistance here. It depends,
> partly, of how much work it would be to wean your code off the Cygwin
> DLL.

Compiling with MinGW or Visual Studio will be (I believe) a lot more work since I have to remove the unixisms from my code which is not insignificant; however, most of this has to do with signals, forks, and spawning. Cygwin seems to have solved most of these porting issues which is why it would be nice if that part worked.

> http://cygwin.com/faq/faq_3.html#SEC102
>
> And when you've sorted it all out, perhaps you could report back to the
> list?

I may try the above faq method; however, I have to do it for all my DLLs and there are seven I use before I get to my DLM. That might also be a lot of work.

Ok, thanks to all who have replied. I believe I got the answer I needed in that "using a Cygwin compiled DLL is impossible with IDL". If I make some progress on this in the form of a generic way to solve this problem, I'll post back. If I just try ifdef'ing out my unix'y code, that might be easier...

Thanks again!

Joey

Subject: Re: Windows IDL DLM with Cygwin
Posted by [Mark Hadfield](#) on Tue, 12 Jul 2005 22:06:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

joey@swri.edu wrote:

> Ok, thanks to all who have replied. I believe I got the answer I needed in
> that "using a Cygwin compiled DLL is impossible with IDL".

The techtip reported by Haje

<http://www.rsinc.com/services/techtip.asp?ttid=3798>

shows that it *is* possible to use Cygwin-compiled DLLs with IDL, via CALL_EXTERNAL. I don't see any mention of the -mno-cygwin switch in the techtip, so it looks like these are true Cygwin DLLs.

Of course DLMs are another matter, but this techtip suggests that maybe you should persist.

A note for those who are completely mystified by this thread:

Cygwin (<http://sources.redhat.com/cygwin/>) is a Unix-emulation system for Windows. The Cygwin package includes a myriad of different tools, one of which is the GCC compiler system, includes the C compiler, gcc. Normally an executable produced by gcc will require the Cygwin DLL at run time. It is this DLL that provides the Unix API functions. It is also possible to use Cygwin's gcc to produce an executable that doesn't require the Cygwin DLL. The command-line switch to accomplish this is "-mno-cygwin".

Mingw (<http://mingw.sourceforge.net/index.shtml>) is a port of the Gnu tools to Windows that doesn't rely on a separate run-time DLL. It does not emulate Unix, so porting Unix software to Mingw is harder than porting it to Cygwin. (The latter is often trivial.) An executable produced by Mingw's gcc will look very much like one produced by Cygwin's gcc with -mno-cygwin. In fact Cygwin bundles its own version of the Mingw headers and import libraries to achieve its "-mno-cygwin" functionality.

--

Mark Hadfield "Kei puwaha te tai nei, Hoea tahi tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: Windows IDL DLM with Cygwin
Posted by [Haje Korth](#) on Wed, 13 Jul 2005 12:11:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mark,
thanks for pointing out to me that the techtip is really for DLLs not DLMs
(but what is the difference of one letter between friends).

To Joey: I would not waste my precious research time to try to get cygwin to compile your code. This is a painful lesson I learned during my times as a graduate student. Even though it is from M\$, I had a lot of luck with Visual Studio. Most code especially research code is generic enough not to rely OS specific extensions, so that I never had to spend much time on conversions.

Cheers,
Haje

"Mark Hadfield" <m.hadfield@niwa.co.nz> wrote in message
news:db1esu\$V4m\$1@newsreader.mailgate.org...
> joey@swri.edu wrote:
>> Ok, thanks to all who have replied. I believe I got the answer I needed
>> in
>> that "using a Cygwin compiled DLL is impossible with IDL".
>
> The techtip reported by Haje
>
> <http://www.rsinc.com/services/techtip.asp?ttid=3798>
>
> shows that it *is* possible to use Cygwin-compiled DLLs with IDL, via
> CALL_EXTERNAL. I don't see any mention of the -mno-cygwin switch in the
> techtip, so it looks like these are true Cygwin DLLs.
>
> Of course DLMs are another matter, but this techtip suggests that maybe
> you should persist.
>
> A note for those who are completely mystified by this thread:
>
> Cygwin (<http://sources.redhat.com/cygwin/>) is a Unix-emulation system for
> Windows. The Cygwin package includes a myriad of different tools, one of
> which is the GCC compiler system, includes the C compiler, gcc. Normally
> an executable produced by gcc will require the Cygwin DLL at run time. It
> is this DLL that provides the Unix API functions. It is also possible to
> use Cygwin's gcc to produce an executable that doesn't require the Cygwin
> DLL. The command-line switch to accomplish this is "-mno-cygwin".
>
> Mingw (<http://mingw.sourceforge.net/index.shtml>) is a port of the Gnu
> tools to Windows that doesn't rely on a separate run-time DLL. It does not
> emulate Unix, so porting Unix software to Mingw is harder than porting it
> to Cygwin. (The latter is often trivial.) An executable produced by
> Mingw's gcc will look very much like one produced by Cygwin's gcc

> with -mno-cygwin. In fact Cygwin bundles its own version of the Mingw
> headers and import libraries to achieve its "-mno-cygwin" functionality.
>
> --
> Mark Hadfield "Kei puwaha te tai nei, Hoesa tahi tatou"
> m.hadfield@niwa.co.nz
> National Institute for Water and Atmospheric Research (NIWA)
