Subject: calling C++ from IDL - throwing exceptions
Posted by Ugo_DiGirolamo on Fri, 09 Feb 2001 20:01:37 GMT
View Forum Message <> Reply to Message

I'm planning to use IDL as a front end for a C++ dll with call_external.
However I couldn't find any way of handling in IDL an exception thrown
from the C++ code.
I was wondering if anyone have an idea about it (also if to confirm
that it's impossible!) or if the DLM way would work better.
However, I've no real idea about what DLM is and I found really little
clue on it in the RSI documentation.
cheers

Ugo Di Girolamo

Sent via Deja.com
http://www.deja.com/

Subject: Re: Calling C++ from IDL
Posted by Nigel Wade on Fri, 16 Nov 2001 09:26:11 GMT
View Forum Message <> Reply to Message

Juan I. Cicuendez wrote:

> Hi,
>
> I am writing you all to see if you could help me because I am
> trying to call C++ code from IDL (I am working in Sun solaris). We
> compile the C++ with CC with extern "C" and produced the .so and when
> we
> use call_external in IDL, it does not find the C++ template library
> which were used in my C++ classes. Is any way to tell IDL where to
> look for this library, or can this be done with make_dll (I don't know
> the compilation options). Could you help me somehow, maybe there are
> other ways, I am a bit desperate. Anyone know the compilation options
> if you want to compile with gcc.
>
> The compilation options I used were:
> CC -c -xarch=v9 otra.c
> CC -G -xarch=v9 otra.o -o libotra.so
>
> from IDL we do:
> S=call_external('libotra.so','mas')
>
>

> Thanks a lot,
>
> Juan Cicuendez

If the library is one of your own then you need to specifically link
against it by adding it to the DSO creation command (CC -G -lyour_lib).

Secondly, so that IDL will know where to look for it (if it's not on the
standard search path) you either need to set the LD_LIBRARY_PATH env
variable to the directory where the library is, or add a "runpath" to the
link command with -R option.

--

Nigel Wade, System Administrator, Space Plasma Physics Group,
University of Leicester, Leicester, LE1 7RH, UK
E-mail : nmw@ion.le.ac.uk
Phone : +44 (0)116 2523568, Fax : +44 (0)116 2523555

Subject: Re: Calling C++ from IDL
Posted by Martin Downing on Fri, 16 Nov 2001 14:35:51 GMT
View Forum Message <> Reply to Message

Hi Jauan,

If Nigel's advice doesnt solve your problem then it may be down to a bug in
the CC compiler.
Many moons ago I reported a bug to Sun where template methods were not being
instantiated when building as a library. The fault was not corrected in the
next few upgrades of solaris, but we are talking 5 years ago now! Note this
had nothing to do with IDL, and if this is the fault you will not be able to
build C++ executables linked to the template libraries either. If this is
the case then maybe they never corrected it, I could dust off the workaround
I came up with.

good luck

Martin

"Nigel Wade" <nmw@ion.le.ac.uk> wrote in message
news:9t2m3j$9oev$1@rook.le.ac.uk...
> Juan I. Cicuendez wrote:
>
>> Hi,
>>
>> I am writing you all to see if you could help me because I am
>> trying to call C++ code from IDL (I am working in Sun solaris). We

>> compile the C++ with CC with extern "C" and produced the .so and when
>> we
>> use call_external in IDL, it does not find the C++ template library
>> which were used in my C++ classes. Is any way to tell IDL where to
>> look for this library, or can this be done with make_dll (I don't know
>> the compilation options). Could you help me somehow, maybe there are
>> other ways, I am a bit desperate. Anyone know the compilation options
>> if you want to compile with gcc.
>>
>> The compilation options I used were:
>> CC -c -xarch=v9 otra.c
>> CC -G -xarch=v9 otra.o -o libotra.so
>>
>> from IDL we do:
>> S=call_external('libotra.so','mas')
>>
>>
>> Thanks a lot,
>>
>> Juan Cicuendez
>
> If the library is one of your own then you need to specifically link
> against it by adding it to the DSO creation command (CC -G -lyour_lib).
>
> Secondly, so that IDL will know where to look for it (if it's not on the
> standard search path) you either need to set the LD_LIBRARY_PATH env
> variable to the directory where the library is, or add a "runpath" to the
> link command with -R option.
>
> --
> ----------
> Nigel Wade, System Administrator, Space Plasma Physics Group,
> University of Leicester, Leicester, LE1 7RH, UK
> E-mail : nmw@ion.le.ac.uk
> Phone : +44 (0)116 2523568, Fax : +44 (0)116 2523555

Subject: Re: Calling C++ from IDL
Posted by jicicuendez on Mon, 19 Nov 2001 09:05:11 GMT
View Forum Message <> Reply to Message

"Martin Downing" <martin.downing@ntlworld.com> wrote in message
news:<RUyJ7.2819$tm3.380540@news11-gui.server.ntli.net>...
> Hi Jauan,
>
> If Nigel's advice doesnt solve your problem then it may be down to a bug in
> the CC compiler.
> Many moons ago I reported a bug to Sun where template methods were not being

> instantiated when building as a library. The fault was not corrected in the
> next few upgrades of solaris, but we are talking 5 years ago now! Note this
> had nothing to do with IDL, and if this is the fault you will not be able to
> build C++ executables linked to the template libraries either. If this is
> the case then maybe they never corrected it, I could dust off the workaround
> I came up with.
>
> good luck
>
> Martin
>
> "Nigel Wade" <nmw@ion.le.ac.uk> wrote in message
> news:9t2m3j$9oev$1@rook.le.ac.uk...
Hi,

I tried Nigel's solution but it does not seem to work. For a very simple class that just prints out stuff with cout, idl says that can't find the cout. When creating the library I added -l with the c++ standard library, I also tried with the -R option giving all the path where the c++ compiler libraries are and it didn't work either. I would appreciate if you could dust off your work around. Even just how to do it with a very simple cout. By the way, I had used the extern "C" in my class.

Many thanks
Juan

>> Juan I. Cicuendez wrote:
>>
>>> Hi,
>>>
>>> I am writing you all to see if you could help me because I am
>>> trying to call C++ code from IDL (I am working in Sun solaris). We
>>> compile the C++ with CC with extern "C" and produced the .so and when
>>> we
>>> use call_external in IDL, it does not find the C++ template library
>>> which were used in my C++ classes. Is any way to tell IDL where to
>>> look for this library, or can this be done with make_dll (I don't know
>>> the compilation options). Could you help me somehow, maybe there are
>>> other ways, I am a bit desperate. Anyone know the compilation options
>>> if you want to compile with gcc.
>>>
>>> The compilation options I used were:
>>> CC -c -xarch=v9 otra.c
>>> CC -G -xarch=v9 otra.o -o libotra.so
>>>
>>> from IDL we do:

>>> S=call_external('libotra.so','mas')
>>>
>>>
>>> Thanks a lot,
>>>
>>> Juan Cicuendez
>>
>> If the library is one of your own then you need to specifically link
>> against it by adding it to the DSO creation command (CC -G -lyour_lib).
>>
>> Secondly, so that IDL will know where to look for it (if it's not on the
>> standard search path) you either need to set the LD_LIBRARY_PATH env
>> variable to the directory where the library is, or add a "runpath" to the
>> link command with -R option.
>>
>> --
>> ----------
>> Nigel Wade, System Administrator, Space Plasma Physics Group,
>> University of Leicester, Leicester, LE1 7RH, UK
>> E-mail : nmw@ion.le.ac.uk
>> Phone : +44 (0)116 2523568, Fax : +44 (0)116 2523555

Subject: Re: Calling C++ from IDL
Posted by Richard Younger on Tue, 20 Nov 2001 16:56:45 GMT
View Forum Message <> Reply to Message

Martin Downing wrote:
>
> Hi Jauan,
>
> If Nigel's advice doesnt solve your problem then it may be down to a
> bug in the CC compiler. Many moons ago I reported a bug to Sun where
> template methods were not being instantiated when building as a
> library. The fault was not corrected in the next few upgrades of
> solaris, but we are talking 5 years ago now! Note this had nothing
> to do with IDL, and if this is the fault you will not be able to
> build C++ executables linked to the template libraries either. If
> this is the case then maybe they never corrected it, I could dust
> off the workaround I came up with.
>
> good luck
>
> Martin

I don't know if this is related to your problems, but as a side note,
the most intensive user of template libraries I know of is the Blitz++
numerics library, which uses obscure (to me, anyway) template features

to gain numerical speed while saving object orientation. They have a website: <http://www.oonumerics.org/blitz>, and they list the solaris compilers as being incompatible due to incomplete template implementation.

A list of template-friendly compilers (but only skimpy discussion) is at <http://www.oonumerics.org/blitz/platforms/>.

Best,

Rich

--
Richard Younger
