Subject: Re: Using C++ DLM's With IDL?
Posted by Nigel Wade on Tue, 31 Jul 2001 14:41:49 GMT
View Forum Message <> Reply to Message

K. Banerjee wrote:

> I need to call some outside code from IDL. I created a
> shared object file, using a C++ compiler (g++). When I call
> a function from the DLM from the IDL command prompt, I get
> the error:
>
> Symbol not found: cerr
>
> (There are other symbols not found.)
>
> Is it possible to use C++ shared objects for DLM's ?
> (I have to use some classes for my routine.)
>
> Thanks.
>
> K. Banerjee
>
>

You don't say what platform you are using, but I'll hazzard a guess that it's Linux.

Did you use g++ to link the DLM? It's generally a wise thing to use the same tool to create a shared object as you use to compile the source.

You could try linking the DLM with g++, or adding the the relevent c++ library to the link command - cerr is in one of the libraries called /usr/lib/libstdc++..., the particular one depends on what version of g++ and libc you are using.

--

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: Using C++ DLM's With IDL?
Posted by K. Banerjee on Tue, 31 Jul 2001 17:18:35 GMT
View Forum Message <> Reply to Message

Nigel Wade <nmw@ion.le.ac.uk> wrote:

> K. Banerjee wrote:

>> I need to call some outside code from IDL. I created a
>> shared object file, using a C++ compiler (g++). When I call
>> a function from the DLM from the IDL command prompt, I get
>> the error:
>>
>> Symbol not found: cerr
>>
>> (There are other symbols not found.)
>>
>> Is it possible to use C++ shared objects for DLM's ?
>> (I have to use some classes for my routine.)
>>
>> Thanks.
>>
>> K. Banerjee
>>
>>

> You don't say what platform you are using, but I'll hazzard a guess that
> it's Linux.

> Did you use g++ to link the DLM? It's generally a wise thing to use the
> same tool to create a shared object as you use to compile the source.

> You could try linking the DLM with g++, or adding the the relevent c++
> library to the link command - cerr is in one of the libraries called
> /usr/lib/libstdc++..., the particular one depends on what version of g++
> and libc you are using.

Yes, I am using g++ on Linux. Do you know off hand how to link
with a shared object library? (libstdc++... is a shared object).

Thanks.

K. Banerjee

Subject: Re: Using C++ DLM's With IDL?
Posted by Nigel Wade on Wed, 01 Aug 2001 08:44:29 GMT
View Forum Message <> Reply to Message

K. Banerjee wrote:

> Nigel Wade <nmw@ion.le.ac.uk> wrote:
>
>> K. Banerjee wrote:

>>
>
>>> I need to call some outside code from IDL. I created a
>>> shared object file, using a C++ compiler (g++). When I call
>>> a function from the DLM from the IDL command prompt, I get
>>> the error:
>>>
>>> Symbol not found: cerr
>>>
>>> (There are other symbols not found.)
>>>
>>> Is it possible to use C++ shared objects for DLM's ?
>>> (I have to use some classes for my routine.)
>>>
>>> Thanks.
>>>
>>> K. Banerjee
>>>
>>>
>>>
>
>> You don't say what platform you are using, but I'll hazzard a guess that
>> it's Linux.
>>
>
>> Did you use g++ to link the DLM? It's generally a wise thing to use the
>> same tool to create a shared object as you use to compile the source.
>>
>
>> You could try linking the DLM with g++, or adding the the relevent c++
>> library to the link command - cerr is in one of the libraries called
>> /usr/lib/libstdc++..., the particular one depends on what version of g++
>> and libc you are using.
>>
>
> Yes, I am using g++ on Linux. Do you know off hand how to link
> with a shared object library? (libstdc++... is a shared object).
>
> Thanks.
>
> K. Banerjee
>

  -l<library_name>, in this case <library_name> is stdc++<whatever>. I
don't know which stdc++ library you'll need; I have 5 in my /usr/lib and
  I haven't a clue what each is for.

a shared object is just a library as far as the linker is concerned.

If you use g++ I would have thought that g++ would add the correct
library to the link command for you, though. Is there a reason you
prefer to use ld rather than g++?

If you really want to use ld, try using g++ first with the -v option so
it lists each command it executes. That should show you which libraries
it uses for the ld command.

--

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: Using C++ DLM's With IDL?
Posted by Craig Markwardt on Wed, 01 Aug 2001 14:38:36 GMT
View Forum Message <> Reply to Message

Nigel Wade <nmw@ion.le.ac.uk> writes:
>> Yes, I am using g++ on Linux. Do you know off hand how to link
>> with a shared object library? (libstdc++... is a shared object).
>>
>> Thanks.
>>
>> K. Banerjee
>>
>
> -l<library_name>, in this case <library_name> is stdc++<whatever>. I
> don't know which stdc++ library you'll need; I have 5 in my /usr/lib and
> I haven't a clue what each is for.
>
> a shared object is just a library as far as the linker is concerned.
>
> If you use g++ I would have thought that g++ would add the correct
> library to the link command for you, though. Is there a reason you
> prefer to use ld rather than g++?
>

Hi Nigel and K.--

I suspect that the C++ runtime system must be initialized before you
can run a C++ module. I am not sure how this is done, and probably it
is rather system dependent. The problem is that IDL is not a C++
program, so this C++ initialization never occurs. I am not sure
whether it is or is not possible to do this at dynamic load time. Or

if, such initialization happens automatically with dynamic loading.

An acceptable alternative may be to have your C++ module be a separate
program, and communicate with IDL using pipes (ie, SPAWN, ...,
UNIT=unit).

Good luck,
Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: Using C++ DLM's With IDL?
Posted by Stein Vidar Hagfors H[1] on Wed, 01 Aug 2001 15:58:05 GMT
View Forum Message <> Reply to Message

Craig Markwardt <craigmnet@cow.physics.wisc.edu> writes:
[..]
> Hi Nigel and K.--
>
> I suspect that the C++ runtime system must be initialized before you
> can run a C++ module. I am not sure how this is done, and probably it
> is rather system dependent. The problem is that IDL is not a C++
> program, so this C++ initialization never occurs. I am not sure
> whether it is or is not possible to do this at dynamic load time. Or
> if, such initialization happens automatically with dynamic loading.
>
> An acceptable alternative may be to have your C++ module be a separate
> program, and communicate with IDL using pipes (ie, SPAWN, ...,
> UNIT=unit).

It may be that the streams like cout/cerr etc need to be initialized,
as well as any static instances of classes, but not much else (?). It
may be possible to rewrite the code to avoid using these features?

--
---------- ----------
Stein Vidar Hagfors Haugan
ESA SOHO SOC/European Space Agency Science Operations Coordinator for SOHO

NASA Goddard Space Flight Center, Email: shaugan@esa.nascom.nasa.gov
Mail Code 682.3, Bld. 26, Room G-1, Tel.: 1-301-286-9028/240-354-6066
Greenbelt, Maryland 20771, USA. Fax: 1-301-286-0264
---------- ----------

Subject: Re: Using C++ DLM's With IDL?
Posted by Nigel Wade on Thu, 02 Aug 2001 15:39:25 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:

> Nigel Wade <nmw@ion.le.ac.uk> writes:
>
>>> Yes, I am using g++ on Linux. Do you know off hand how to link
>>> with a shared object library? (libstdc++... is a shared object).
>>>
>>> Thanks.
>>>
>>> K. Banerjee
>>>
>>>
>> -l<library_name>, in this case <library_name> is stdc++<whatever>. I
>> don't know which stdc++ library you'll need; I have 5 in my /usr/lib and
>> I haven't a clue what each is for.
>>
>> a shared object is just a library as far as the linker is concerned.
>>
>> If you use g++ I would have thought that g++ would add the correct
>> library to the link command for you, though. Is there a reason you
>> prefer to use ld rather than g++?
>>
>>
>
> Hi Nigel and K.--
>
> I suspect that the C++ runtime system must be initialized before you
> can run a C++ module. I am not sure how this is done, and probably it
> is rather system dependent. The problem is that IDL is not a C++
> program, so this C++ initialization never occurs. I am not sure
> whether it is or is not possible to do this at dynamic load time. Or
> if, such initialization happens automatically with dynamic loading.
>
> An acceptable alternative may be to have your C++ module be a separate
> program, and communicate with IDL using pipes (ie, SPAWN, ...,
> UNIT=unit).
>
> Good luck,
> Craig
>
>

I'm not exactly sure what a "C++ runtime system" might consist of...

Anyway, I've managed to create a simple DLM from C++ source and run it

in IDL 5.3. The core code is in C++ (I borrowed it from an example mex file for MATLAB), the DLM interface code is in C. The C++ was compiled with g++, the C code with gcc. Both were linked into a DSO with g++.

Pretty straight forward, and it works just like it does for C and FORTRAN.
