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.
