Subject: Re: g++ with IDL call_external
Posted by wem on Mon, 10 Oct 2005 20:25:56 GMT
View Forum Message <> Reply to Message

g++ is the traditional nickname of GNU C++, a freely redistributable C++ compiler. It is part of gcc, the GNU compiler suite, and is currently part of that distribution.
(http://directory.fsf.org/GNU/gpp.html)

thought g++ calls gcc or vice versa, but that they should have the same output ...

Denis Barkats wrote:
> Hi, I've been trying to use call_external from within IDL to call a
> piece of code that I compiled with g++ instead of gcc. Here is the
> exemple.
>
> i have a file simple.c:
>
> #include <stdio.h>
> #include <stdlib.h>
>
> int Simple() {
> return 13;
> }
>
> Using gcc I do
> -gcc -c simple.c
> -gcc -bundle -flat_namespace -o simple.so simple.o
> which creates the sharable librairie simple.so which I can call from
> IDL as
> -print,call_external('/Users/denis/idl/bicep/simple.so','Sim ple',
> /unload)
> 13
>
> So everything is fine.
> However, now if I use g++
>
> g++ -c simple.c
> g++ -bundle -flat_namespace -o simple.so simple.o
> print,call_external('/Users/denis/idl/bicep/simple.so','Simp le',
> /unload)
> % CALL_EXTERNAL: Error loading sharable executable.
> Symbol: Simple, File =
> /Users/denis/idl/bicep/simple.so
> symbol not found
> it does not seem to work.
>

> DO any of you g++ users have any idea ?
> Thanks
>

Subject: Re: g++ with IDL call_external
Posted by Nigel Wade on Tue, 11 Oct 2005 12:07:52 GMT
View Forum Message <> Reply to Message

Denis Barkats wrote:

> Hi, I've been trying to use call_external from within IDL to call a
> piece of code that I compiled with g++ instead of gcc. Here is the
> exemple.
>
> i have a file simple.c:
>
> #include <stdio.h>
> #include <stdlib.h>
>
> int Simple() {
> return 13;
> }
>
> Using gcc I do
> -gcc -c simple.c
> -gcc -bundle -flat_namespace -o simple.so simple.o
> which creates the sharable librairie simple.so which I can call from
> IDL as
> -print,call_external('/Users/denis/idl/bicep/simple.so','Sim ple',
> /unload)
> 13
>
> So everything is fine.
> However, now if I use g++
>
> g++ -c simple.c
> g++ -bundle -flat_namespace -o simple.so simple.o
> print,call_external('/Users/denis/idl/bicep/simple.so','Simp le',
> /unload)
> % CALL_EXTERNAL: Error loading sharable executable.
> Symbol: Simple, File =
> /Users/denis/idl/bicep/simple.so
> symbol not found
> it does not seem to work.
>
> DO any of you g++ users have any idea ?
> Thanks

My guess would be that you need to disable the g++ name-mangling. If you look at the names produced in simple.o by g++ you will see that Simple has been mangled into something strange like _X6Simplev (the mangled name includes a signature, to allow overloading). You need to declare Simple in an extern 'C', to tell g++ that it's a C function, not a C++ one, so overloading and mangling can be disabled.

I'll leave it as an exercise for the OP to determine how to do this (principally because I can't remember the exact semantics).

--
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 2523548, Fax : +44 (0)116 2523555

Subject: Re: g++ with IDL call_external
Posted by Denis Barkats on Tue, 11 Oct 2005 19:22:08 GMT
View Forum Message <> Reply to Message

Thanks Nigel,

that is exactly it
if I nm the simple.o I see there is a function called _Z6simplev and I can call that one.

Thanks

Subject: Re: g++ with IDL call_external
Posted by tdaitx on Wed, 12 Oct 2005 02:05:54 GMT
View Forum Message <> Reply to Message

As Nigel said, C++ compilers mangle functions name differently than C compilers, since they have to support function overloading. You should use

```
#ifdef __cplusplus
extern "C" {
#endif
 <all your C prototype code goes here>
#ifdef __cplusplus
}
```

#endif

and you'll be able to compile it with both gcc and g++.

You should really take a look at
http://developers.sun.com/prodtech/cc/articles/mixing.html and
http://www.parashift.com/c++-faq-lite/mixing-c-and-cpp.html, both an excellent source on how to mix C and C++ code.

Regards,
Tiago S Daitx
