Subject: C++ in IDL
Posted by chrisduckworth on Fri, 22 Feb 2002 01:25:11 GMT
View Forum Message <> Reply to Message

When I try to build DLL files with the export.h file in IDL
5.5 my C++ compiler gives me an error on line 1864 (~). I have tried
this with two compilers codewarrior and borland. Is there somthing
that I am doing wrong?

Also, does anyone have a detailed walkthrough on creating DLM's for
IDL. IDL's documentation seems a little light.

Thanks,
CND

Subject: Re: C++ in IDL
Posted by Antonio Santiago on Tue, 31 May 2005 06:37:31 GMT
View Forum Message <> Reply to Message

Tim-Oliver Husser wrote:
> Hi,
>
> this is my first posting here, so let me just say hello. :-)
>
> I've got some small problem with IDL 6.0 and it would give me great
> pleasure, if someone could help me with that.
> There is some heavy computation I have to do within IDL (running a
> genetic algorithm, so it's really pure number crunching) and I was
> thinking about porting that algorithm to C/C++ to call it via
> call_external. I assume, that the code would be much faster in C/C++,
> wouldn't it?
> Now my problem: I do not really want to use ANSI C (I never ever want
> to use malloc again), and anyway C++ is my favorite programming
> language. But I cannot compile C++ code with make_dll and my
> experience in Linux programming is not that good (I'm just a windows
> programmer), so I do not know, how to compile the C++ code correctly.
> I tried something like this:
>
> g++ -fPIC -I"path/to/idl/includes" -L"path/to/idl/libs" -lidl
> -D_REENTRANT test.cpp
>
> But there seem to be missing a lot of libraries. I think that I can
> figure out by myself, which libraries I have to link to, but I do not
> really want to do this, before I know for sure, that it is really
> possible to call C++ libraries with IDL. I just don't want to do some
> useless work...
> All the C++ code is enclosed in an 'extern "C" { }', so that should

> not be a problem I think.
> Has anyone ever tried to do something like this and could help me?
>
> thx,
> Tim-Oliver

Hi I don't know almost anything using call_external and make_dll, right now I only try to modify an example on using WIDGET_STUB and extending IDL widgets.

What libraries seems to missing you? Perhaps you don't have installed correctly g++. What distribution do you use? depends on it, the necessary packages can differ. For example you can have installed the libraries to execute c++ programs but not have installed the headers to compile c++ programs.

Bye.
--

Antonio Santiago Pï¿½rez
(email: santiago<<at>>grahi.upc.edu)
(www: http://www.grahi.upc.edu/santiago)
(www: http://asantiago.blogsite.org)

GRAHI - Grup de Recerca Aplicada en Hidrometeorologia
Universitat Politï¿½cnica de Catalunya

Subject: Re: C++ in IDL
Posted by Rick Towler on Tue, 31 May 2005 16:35:16 GMT

View Forum Message <> Reply to Message

Tim-Oliver Husser wrote:

> this is my first posting here, so let me just say hello. :-)

Well hello there.

> I've got some small problem with IDL 6.0 and it would give me great
> pleasure, if someone could help me with that.
> There is some heavy computation I have to do within IDL (running a
> genetic algorithm, so it's really pure number crunching) and I was
> thinking about porting that algorithm to C/C++ to call it via
> call_external. I assume, that the code would be much faster in C/C++,
> wouldn't it?

It depends, but most likely yes. I would write a dlm, not use call external.

> Has anyone ever tried to do something like this and could help me?

Sure. I wasn't sure I could write a C++ dlm so I wrote a simple test
program, got that to work, and went from there. I can't comment on your
linux particulars, but I can offer up this simple example which may or
may not be of help. At any rate, this can be done so push on. The
example files:

http://www.acoustics.washington.edu/~towler/programs/Mgc_vec tor.zip

Also, since I wrote this Ronn Kling has updated his book "Calling C from
IDL" to include a chapter on calling C++. I would highly recommend
picking this book up from his website http://www.kilvarock.com.

-Rick

Subject: Re: C++ in IDL
Posted by Tim-Oliver Husser on Wed, 01 Jun 2005 12:33:04 GMT
View Forum Message <> Reply to Message

Rick Towler wrote:
>> I assume, that the code would be much faster in C/C++,
>> wouldn't it?
>
>
> It depends, but most likely yes. I would write a dlm, not use call
> external.

I've just converted the fitness function to C and it was faster than the
IDL version about the factor 100. I think IDL hasn't really been
designed for number crunching... ;)

> Sure. I wasn't sure I could write a C++ dlm so I wrote a simple test
> program, got that to work, and went from there. I can't comment on your
> linux particulars, but I can offer up this simple example which may or
> may not be of help. At any rate, this can be done so push on. The
> example files:
>
> http://www.acoustics.washington.edu/~towler/programs/Mgc_vec tor.zip

Thanks, I'll have a look at it.

> Also, since I wrote this Ronn Kling has updated his book "Calling C from

> IDL" to include a chapter on calling C++. I would highly recommend
> picking this book up from his website http://www.kilvarock.com.

Hmm, looks nice, I hope my boss will buy this for me... ;)

Tim-Oliver

Subject: Re: C++ in IDL
Posted by Michael Wallace on Wed, 01 Jun 2005 15:31:16 GMT
View Forum Message <> Reply to Message

> I've just converted the fitness function to C and it was faster than the
> IDL version about the factor 100. I think IDL hasn't really been
> designed for number crunching... ;)

There's that much of a difference? I thought sure that IDL would have
been designed for number crunching considering the market they are
trying to serve. Of course, if the IDL code wasn't optimized...

-Mike

Subject: Re: C++ in IDL
Posted by David Fanning on Wed, 01 Jun 2005 16:10:34 GMT
View Forum Message <> Reply to Message

Michael Wallace writes:

> There's that much of a difference? I thought sure that IDL would have
> been designed for number crunching considering the market they are
> trying to serve. Of course, if the IDL code wasn't optimized...

I've seen this kind of comparison many times. On further
investigation the IDL code is usually, uh, not well written. :-)
I would take any comparisons not performed by J.D. with a
grain of salt. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: C++ in IDL
Posted by JD Smith on Wed, 01 Jun 2005 17:00:34 GMT
View Forum Message <> Reply to Message

On Wed, 01 Jun 2005 10:10:34 -0600, David Fanning wrote:

> Michael Wallace writes:
>
>> There's that much of a difference? I thought sure that IDL would have
>> been designed for number crunching considering the market they are
>> trying to serve. Of course, if the IDL code wasn't optimized...
>
> I've seen this kind of comparison many times. On further
> investigation the IDL code is usually, uh, not well written. :-)
> I would take any comparisons not performed by J.D. with a
> grain of salt. :-)

I've been known to push on IDL to get as much performance as possible, and still, for many types of problems, a direct C-coded approach, compiled with optimization, will out-perform the best-optimized IDL version by a factor of 10 or more. Here's a good example from your site in which we collected and refined the best-optimized IDL algorithms from the experts (Craig, Wayne, etc.), and still I found a simple compiled C approach is 20x faster:

http://www.dfanning.com/code_tips/drizzling.html

IDL truly excels at basic large array operations (adding 1 to 1 million integers, etc.); for these types of problems you'll find comparable performance to compiled code (ballpark anyway). For more complex algorithms, you can make huge gains in performance with careful vectorization and other methods in IDL, but, in the end, may still (but not always) miss compiled performance by a non-negligible margin.

This is not actually surprising. It's part of the deal you make when you code in IDL at a high-level, vs., e.g., C or FORTRAN at a low level. Did you think you would get all of those many conveniences which IDL provides, like loose typing, dynamic variable content, optional and keyword arguments, variable type checking, etc., for free? A much fairer comparison would be between, e.g., Matlab's array-crunching performance and IDL's. Among the so-called 4G languages (including scripting extensions like Perl's PDL, etc.), IDL delivers among the best performance for basic array operations.

I might suggest prototyping everything in IDL, profiling to find the algorithm or sub-algorithm which is limiting performance, and then re-coding that in C as a DLM. Sometimes, as in the case referenced above, the C version can be quite simple and compact, especially if

all of the setup, display, analysis and data juggling is handled in
IDL. Simple DLMs are actually not difficult to produce, and it's
worth your while to learn how to make use of them. Their one real
drawback is cross-platform compatibility, but with MAKE_DLL and auto
glue, you can actually recover this to some degree.

JD

Subject: Re: C++ in IDL
Posted by R.G. Stockwell on Wed, 01 Jun 2005 22:42:50 GMT
View Forum Message <> Reply to Message

"David Fanning" <davidf@dfanning.com> wrote in message
news:MPG.1d078b3e7c082ef4989691@news.frii.com...
...
> I would take any comparisons not performed by J.D. with a
> grain of salt. :-)
>
> Cheers,
>
> David

Right on! I know of people very nearby that have concluded that
IDL is too slow, and they have to convert code to C. That may be case,
but I bet a careful eye might make IDL less slow.

In cases like that I always find myself asking "What would J.D. do?"

Cheers,
bob

--
WWJDD?

Subject: Re: C++ in IDL
Posted by Rick Towler on Thu, 02 Jun 2005 16:32:48 GMT
View Forum Message <> Reply to Message

R.G. Stockwell wrote:

> WWJDD?

:)

Who's printing up the bumper stickers and t-shirts?

-Rick

Come on RSI. Have some fun with this. Print up some IDL shirts with WWJDD on the back and give them out as prizes. Throw us a bone, we're the best marketeers you have. Foster and grow the community...