
Subject: Re: IDL --> C++ ?

Posted by [Pavel A. Romashkin](#) on Thu, 24 Oct 2002 21:14:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

Robert,

I envy your ambitious mind and ample time at hand.

I am sure the NG will be anxiously waiting for your program. I am afraid

I will not see it as I plan to retire in some 30 years.

Good luck,

Pavel

Robert wrote:

>

> Hi folks,

>

> is there any application out there to translate IDL source code to C++

> source code ? Even buggy solutions are welcome ,) ...

> If there is no possibility right now, i'm going to programm eigher a full

> freatured (free !) translator or a set of classes/ librarys that'll allow

> easy rewriting any IDL application within C++ . (helping hands and minds

> are more then welcome...)

>

> thank you for reading and for any response,

>

> Robert

Subject: Re: IDL --> C++ ?

Posted by [Robert\[1\]](#) on Thu, 24 Oct 2002 21:58:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

Pavel A. Romashkin wrote:

> Robert,

> I envy your ambitious mind and ample time at hand.

> I am sure the NG will be anxiously waiting for your program. I am afraid

> I will not see it as I plan to retire in some 30 years.

> Good luck,

> Pavel

>

Hey hi Pavel,

how long do you need to write a "Hello World" ? more or less then a year ?

*g , but no kidding, some work is allready done and I'm sure to write the
core funktions woun't took 30 years or more ,)

By the way , you're answer has not been verry helpful. Instead of trying to
mess around with me you shall better type something useful.

cheers,
Robert

Subject: Re: IDL --> C++ ?

Posted by [Mark Hadfield](#) on Thu, 24 Oct 2002 22:24:22 GMT

[View Forum Message](#) <> [Reply to Message](#)

"Robert" <Rober-Mueller@My-Mail.ch> wrote in message
news:ap9qfr\$act\$1@news.online.de...

> Pavel A. Romashkin wrote:

>

>> I envy your ambitious mind and ample time at hand. I am sure the

>> NG will be anxiously waiting for your program. I am afraid I will

>> not see it as I plan to retire in some 30 years.

>

> By the way , you're answer has not been verry helpful. Instead of

> trying to mess around with me you shall better type something

> useful.

I think you would do well to get used to the style of this newsgroup,
Robert. Flaming is definitely not appreciated but gently barbed humour
(as in Pavel's message) is, and is considered to be something of an
art form. Please do not take it personally.

I think Pavel was being very helpful. He was implying that writing a
useful IDL -> C++ syntax translator would be a huge job. Don't go
there!

One the other hand, pulling off this feat should earn you a permanent
membership on the IDL Expert Programmers Association.

--

Mark Hadfield "Ka puwaha te tai nei, Hoesa tatou"

m.hadfield@niwa.co.nz

National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: IDL --> C++ ?

Posted by [Justin\[2\]](#) on Thu, 24 Oct 2002 22:32:35 GMT

[View Forum Message](#) <> [Reply to Message](#)

Robert,

I think this might be a near impossible task for you unless you amass an
army to help, or unless you plan to implement only the most basic

implimentation.

Start with plotting. How do you plan to produce plots in C++ that will work across different platforms? Even if you just implemented the screen devices ('Win', 'X', 'MAC') and the 'PS' device, there would be hell of a lot of coding needed to create a plot that looks the same as an IDL plot. If you're considering things like widgets and pointers and object graphics too... well I think it would be tough.

Also if you plan to do a 'simple' translation (i.e. IDL:"a=1" to C++:"a=1;") without complicated run-time handling of variables, how will you handle variable type (C++ is strongly typed, IDL weakly typed)? For instance, in IDL you could happily write:

```
a = '123'  
a = FLOAT(a)
```

and 'a' will change from string to float as needed. A simple translation of this would fail in C++ since a variable cannot change type at run-time. Similarly what about run-time creation of structures (CREATE_STRUCT), for instance, or calling of statements created at run-time (EXECUTE)? I think this list is endless...

Now this wouldn't be impossible (I guess IDL is written in C) but I think it would be mighty hard. I don't think a translator is viable, and to write an interpreter would be duplicating RSI's effort!

Regards,

Justin

Subject: Re: IDL --> C++ ?

Posted by [Justin\[2\]](#) on Thu, 24 Oct 2002 22:33:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

Robert,

I think this might be a near impossible task for you unless you amass an army to help, or unless you plan to implement only the most basic implimentation.

Start with plotting. How do you plan to produce plots in C++ that will work across different platforms? Even if you just implemented the screen devices ('Win', 'X', 'MAC') and the 'PS' device, there would be hell of a lot of coding needed to create a plot that looks the same as an IDL plot. If you're considering things like widgets and pointers and object graphics too... well I think it would be tough.

Also if you plan to do a 'simple' translation (i.e. IDL:"a=1" to C++:"a=1;") without complicated run-time handling of variables, how will you handle variable type (C++ is strongly typed, IDL weakly typed)? For instance, in IDL you could happily write:

```
a = '123'  
a = FLOAT(a)
```

and 'a' will change from string to float as needed. A simple translation of this would fail in C++ since a variable cannot change type at run-time. Similarly what about run-time creation of structures (CREATE_STRUCT), for instance, or calling of statements created at run-time (EXECUTE)? I think this list is endless...

Now this wouldn't be impossible (I guess IDL is written in C) but I think it would be mighty hard. I don't think a translator is viable, and to write an interpreter would be duplicating RSI's effort!

Regards,

Justin

Subject: Re: IDL --> C++ ?

Posted by [Justin\[2\]](#) on Thu, 24 Oct 2002 22:40:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

Robert,

I think this might be a near impossible task for you unless you amass an army to help, or unless you plan to implement only the most basic implimentation.

Start with plotting. How do you plan to produce plots in C++ that will work across different platforms? Even if you just implemented the screen devices ('Win', 'X', 'MAC') and the 'PS' device, there would be hell of a lot of coding needed to create a plot that looks the same as an IDL plot. If you're considering things like widgets and pointers and object graphics too... well I think it would be tough.

Also if you plan to do a 'simple' translation (i.e. IDL:"a=1" to C++:"a=1;") without complicated run-time handling of variables, how will you handle variable type (C++ is strongly typed, IDL weakly typed)? For instance, in IDL you could happily write:

```
a = '123'  
a = FLOAT(a)
```

and 'a' will change from string to float as needed. A simple translation of this would fail in C++ since a variable cannot change type at

run-time. Similarly what about run-time creation of structures (CREATE_STRUCT), for instance, or calling of statements created at run-time (EXECUTE)? I think this list is endless...

Now this wouldn't be impossible (I guess IDL is written in C) but I think it would be mighty hard. I don't think a translator is viable, and to write an interpreter would be duplicating RSI's effort!

Regards,

Justin

Subject: Re: IDL --> C++ ?

Posted by [Justin\[2\]](#) on Thu, 24 Oct 2002 22:40:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

Robert,

I think this might be a near impossible task for you unless you amass an army to help, or unless you plan to implement only the most basic implimentation.

Start with plotting. How do you plan to produce plots in C++ that will work across different platforms? Even if you just implemented the screen devices ('Win', 'X', 'MAC') and the 'PS' device, there would be hell of a lot of coding needed to create a plot that looks the same as an IDL plot. If you're considering things like widgets and pointers and object graphics too... well I think it would be tough.

Also if you plan to do a 'simple' translation (i.e. IDL:"a=1" to C++:"a=1;") without complicated run-time handling of variables, how will you handle variable type (C++ is strongly typed, IDL weakly typed)? For instance, in IDL you could happily write:

```
a = '123'  
a = FLOAT(a)
```

and 'a' will change from string to float as needed. A simple translation of this would fail in C++ since a variable cannot change type at run-time. Similarly what about run-time creation of structures (CREATE_STRUCT), for instance, or calling of statements created at run-time (EXECUTE)? I think this list is endless...

Now this wouldn't be impossible (I guess IDL is written in C) but I think it would be mighty hard. I don't think a translator is viable, and to write an interpreter would be duplicating RSI's effort!

Regards,

Justin

Subject: Re: IDL --> C++ ?

Posted by [Justin\[2\]](#) on Thu, 24 Oct 2002 22:42:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

Robert,

I think this might be a near impossible task for you unless you amass an army to help, or unless you plan to implement only the most basic implimentation.

Start with plotting. How do you plan to produce plots in C++ that will work across different platforms? Even if you just implemented the screen devices ('Win', 'X', 'MAC') and the 'PS' device, there would be hell of a lot of coding needed to create a plot that looks the same as an IDL plot. If you're considering things like widgets and pointers and object graphics too... well I think it would be tough.

Also if you plan to do a 'simple' translation (i.e. IDL:"a=1" to C++:"a=1;") without complicated run-time handling of variables, how will you handle variable type (C++ is strongly typed, IDL weakly typed)? For instance, in IDL you could happily write:

```
a = '123'
```

```
a = FLOAT(a)
```

and 'a' will change from string to float as needed. A simple translation of this would fail in C++ since a variable cannot change type at run-time. Similarly what about run-time creation of structures (CREATE_STRUCT), for instance, or calling of statements created at run-time (EXECUTE)? I think this list is endless...

Now this wouldn't be impossible (I guess IDL is written in C) but I think it would be mighty hard. I don't think a translator is viable, and to write an interpreter would be duplicating RSI's effort!

Regards,

Justin

Subject: Re: IDL --> C++ ?

Posted by [Justin\[2\]](#) on Thu, 24 Oct 2002 22:42:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

Robert,

I think this might be a near impossible task for you unless you amass an army to help, or unless you plan to implement only the most basic implimentation.

Start with plotting. How do you plan to produce plots in C++ that will work across different platforms? Even if you just implemented the screen devices ('Win', 'X', 'MAC') and the 'PS' device, there would be hell of a lot of coding needed to create a plot that looks the same as an IDL plot. If you're considering things like widgets and pointers and object graphics too... well I think it would be tough.

Also if you plan to do a 'simple' translation (i.e. IDL:"a=1" to C++:"a=1;") without complicated run-time handling of variables, how will you handle variable type (C++ is strongly typed, IDL weakly typed)? For instance, in IDL you could happily write:

```
a = '123'  
a = FLOAT(a)
```

and 'a' will change from string to float as needed. A simple translation of this would fail in C++ since a variable cannot change type at run-time. Similarly what about run-time creation of structures (CREATE_STRUCT), for instance, or calling of statements created at run-time (EXECUTE)? I think this list is endless...

Now this wouldn't be impossible (I guess IDL is written in C) but I think it would be mighty hard. I don't think a translator is viable, and to write an interpreter would be duplicating RSI's effort!

Regards,

Justin

Subject: Re: IDL --> C++ ?

Posted by [Justin\[2\]](#) on Thu, 24 Oct 2002 22:45:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Robert,

I think this might be a near impossible task for you unless you amass an army to help, or unless you plan to implement only the most basic implimentation.

Start with plotting. How do you plan to produce plots in C++ that will work across different platforms? Even if you just implemented the screen devices ('Win', 'X', 'MAC') and the 'PS' device, there would be hell of

a lot of coding needed to create a plot that looks the same as an IDL plot. If you're considering things like widgets and pointers and object graphics too... well I think it would be tough.

Also if you plan to do a 'simple' translation (i.e. IDL:"a=1" to C++:"a=1;") without complicated run-time handling of variables, how will you handle variable type (C++ is strongly typed, IDL weakly typed)? For instance, in IDL you could happily write:

```
a = '123'  
a = FLOAT(a)
```

and 'a' will change from string to float as needed. A simple translation of this would fail in C++ since a variable cannot change type at run-time. Similarly what about run-time creation of structures (CREATE_STRUCT), for instance, or calling of statements created at run-time (EXECUTE)? I think this list is endless...

Now this wouldn't be impossible (I guess IDL is written in C) but I think it would be mighty hard. I don't think a translator is viable, and to write an interpreter would be duplicating RSI's effort!

Regards,

Justin

Subject: Re: IDL --> C++ ?

Posted by [Justin\[2\]](#) on Thu, 24 Oct 2002 22:45:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

Robert,

I think this might be a near impossible task for you unless you amass an army to help, or unless you plan to implement only the most basic implimentation.

Start with plotting. How do you plan to produce plots in C++ that will work across different platforms? Even if you just implemented the screen devices ('Win', 'X', 'MAC') and the 'PS' device, there would be hell of a lot of coding needed to create a plot that looks the same as an IDL plot. If you're considering things like widgets and pointers and object graphics too... well I think it would be tough.

Also if you plan to do a 'simple' translation (i.e. IDL:"a=1" to C++:"a=1;") without complicated run-time handling of variables, how will you handle variable type (C++ is strongly typed, IDL weakly typed)? For instance, in IDL you could happily write:

```
a = '123'
```


a = FLOAT(a)

and 'a' will change from string to float as needed. A simple translation of this would fail in C++ since a variable cannot change type at run-time. Similarly what about run-time creation of structures (CREATE_STRUCT), for instance, or calling of statements created at run-time (EXECUTE)? I think this list is endless...

Now this wouldn't be impossible (I guess IDL is written in C) but I think it would be mighty hard. I don't think a translator is viable, and to write an interpreter would be duplicating RSI's effort!

Regards,

Justin

Subject: Re: IDL --> C++ ?

Posted by [Kenneth P. Bowman](#) on Thu, 24 Oct 2002 23:55:35 GMT

[View Forum Message](#) <> [Reply to Message](#)

In article <ap9kjj\$7cr\$3@news.online.de>,
Robert <Rober-Mueller@My-Mail.ch> wrote:

> Hi folks,
>
> is there any application out there to translate IDL source code to C++
> source code ? Even buggy solutions are welcome ,) ...
> If there is no possibility right now, i'm going to programm eigher a full
> freatured (free !) translator or a set of classes/ librarys that'll allow
> easy rewriteing any IDL application within C++ . (helping hands and minds
> are more then welcome...)

Are you by any chance referring to Interface Definition Language (IDL), rather than Interactive Data Language (er, IDL)? This newsgroup is the latter. The former has its own newsgroup: comp.lang.idl.

Cheers, Ken Bowman

Subject: Re: IDL --> C++ ?

Posted by [Justin\[2\]](#) on Fri, 25 Oct 2002 19:26:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

Oops. Looks like I posted this message a few more times than I had meant to. Sorry about that. Still it was a very good post, so it probably deserved it...

Subject: Re: IDL --> C++ ?

Posted by [muswick](#) on Sun, 27 Oct 2002 04:04:12 GMT

[View Forum Message](#) <> [Reply to Message](#)

Your question has been asked several times over the years, (maybe not to c++, but a translator to some other language.)

I think you need to ask yourself why you want the translator. I can think of several, but the big one is:

Once C/C++ is code is made, the results could be compiled and you could either sell or give your application away, without others needing to buy runtime licenses or you having to pay royalty fees to RSI. Given the approximate \$500 cost of runtime licenses, this would be desirable for simple applications. But if you have something complex that is worth selling, than you basically have two choices, rewrite the application, or pass the cost of the license onto your customers. My company, chose the latter, since staying with IDL has advantages for us.

For simple apps, or research groups, etc. you're using IDL because of it's power, capabilities and ease of prototyping code, interactive testing to application. That power and flexibility is what RSI sells, and we pay for. A lot of work has and continues to go into IDL, and RSI, needs to pay it's staff and make a profit. For groups, or company internal use, there are a variety of license methods available to fit needs and budgets.

If you ever could write such a translator, it would be worth selling, or charge people for the service of doing the translation, maybe 10 cents per line of IDL code. But keep in mind, that IDL has long history and is pretty much backward compatible. I am working on a large application written some time ago, where the original programmer implemented objects using HANDLES. The code still runs fine under IDL 5.5, but I had to dust cobwebs from my brain cells to remember how to use them. Those are just some of the obsolete stuff you'd have include.

Maybe posting why you want the translator, might help someone in the newsgroup give you some alternatives.

Good luck

Gary

Robert <Rober-Mueller@My-Mail.ch> wrote in message
news:<ap9kh3\$7cr\$1@news.online.de>...

> Hi folks,

>
> is there any application out there to translate IDL source code to C++
> source code ? Even buggy solutions are welcome ,) ...
> If there is no possibility right now, i'm going to programm either a full
> featured (free !) translator or a set of classes/ librarys that'll allow
> easy rewriteing any IDL application within C++ . (helping hands and minds
> are more then welcome...)
>
> thank you for reading and for any response,
>
> Stefan
