
Subject: C/C++ conversion to IDL

Posted by [nasalmon](#) on Sun, 29 May 2005 10:13:06 GMT

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

Would anybody know if there is there software available to convert programs in C into IDL (i know you can link in C programs to running under IDL)? However, i am more interested in getting shorter, more transparent code which is easier to work with. I obviously want the minimum of hassel in converting the C code into IDL, so i dont want to trawl through this myself converting it.

Thank you,
Neil

Subject: Re: C/C++ conversion to IDL

Posted by [marc schellens\[1\]](#) on Mon, 06 Jun 2005 09:43:40 GMT

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

For simple routines: Use C.

Learning C++ takes longer (learning C++ includes learning (most of) C).

C++ would be the better choice if you expect your programs to get complex.

just my 2 cent,
marc

Subject: Re: C/C++ conversion to IDL

Posted by [nasalmon](#) on Mon, 06 Jun 2005 15:13:02 GMT

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

Just searching for a free C / C++ compiler, are there any that anyone could recommend that might work well under Windows XP and enabling calling object files from and interfacing with IDL? Might it even be possible to run / comple C / C++ from with inside IDL using the IDL editor?

Free compilers i heard about are DJGPP, GNU, Borland and Dev-C++, would any of these be suitable or preferable?

Cheers,
Neil

Subject: Re: C/C++ conversion to IDL

Posted by [Haje Korth](#) on Mon, 06 Jun 2005 15:53:03 GMT

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

I have never gotten the gnu compiler to work with IDL despite hard trying so I went to M\$ C. However, in the meantime there is a techtip on the RSI website to get it working with cygwin version of gcc. Give that a shot.

Haje

<nasalmon@onetel.net.uk> wrote in message

news:1118070782.438499.233390@o13g2000cwo.googlegroups.com.. .

- > Just searching for a free C / C++ compiler, are there any that anyone
- > could recommend that might work well under Windows XP and enabling
- > calling object files from and interfacing with IDL? Might it even be
- > possible to run / comple C / C++ from with inside IDL using the IDL
- > editor?
- >
- > Free compilers i heard about are DJGPP, GNU, Borland and Dev-C++, would
- > any of these be suitable or preferable?
- >
- > Cheers,
- > Neil
- >

Subject: Re: C/C++ conversion to IDL

Posted by [Karl Schultz](#) on Mon, 06 Jun 2005 15:53:30 GMT

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

On Mon, 06 Jun 2005 08:13:02 -0700, nasalmon wrote:

- > Just searching for a free C / C++ compiler, are there any that anyone
- > could recommend that might work well under Windows XP and enabling
- > calling object files from and interfacing with IDL? Might it even be
- > possible to run / comple C / C++ from with inside IDL using the IDL
- > editor?
- >
- > Free compilers i heard about are DJGPP, GNU, Borland and Dev-C++, would
- > any of these be suitable or preferable?

I have not actually tried it (yet), but this looks promising:

<http://msdn.microsoft.com/visualc/vctoolkit2003/>

You can probably use IDL's MAKE_DLL procedure and this compiler to build your C/C++ code from "inside IDL". And I suppose that you could use IDL's editor as well, but if you are going to do a lot of C code, you would want

to use emacs or something with more C language syntax support.

Karl

Subject: Re: C/C++ conversion to IDL

Posted by [Michael Wallace](#) on Mon, 06 Jun 2005 16:06:56 GMT

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

- > Would there be any preference for C or C++, as i am not really that
- > familiar with either and dont really have the time to learn one and
- > realise later that the other one is more appropriate?

C++ is almost, but not quite, a superset of C. Because of this, I'd advise learning C first and then go on to learn C++ should you need to.

The primary difference between the languages is that C++ supports object-oriented programming while C is procedural. If you won't be using objects, don't worry about learning C++ now. There are other differences between the languages, but none of those make a compelling case to learn C++ in this situation.

-Mike

Subject: Re: C/C++ conversion to IDL

Posted by [Rick Towler](#) on Mon, 06 Jun 2005 17:25:29 GMT

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

- > Basically i wanted to increase the speed of my IDL programs by calling
- > certain time consuming modules compiled in C or C++. I do a lot of
- > calling of routines recursively and working with binary trees.

You'll want to get Ronn Kling's "Calling C from IDL" (which now includes a chapter on C++). I have found this to be a great introduction to creating .dlms in IDL. www.kilvarock.com (it is "unavailable" from Amazon but hopefully Ronn still has copies)

Some good suggestions were given regarding a compiler. As for editors, I like jEdit (www.jedit.org). It's an editor, not an IDE, but a fine editor indeed.

- > Would there be any preference for C or C++, as i am not really that
- > familiar with either and dont really have the time to learn one and
- > realise later that the other one is more appropriate?

I would look for appropriate examples in both C/C++ and go from there.

For example, if you can find an implementation of your data structure in C that gives you 90% of what you want then go with that code in C. Or if you find some C++ code that is just what you want, then C++. This isn't about learning a new language, it's about solving your problem quickly. IMO.

-Rick

Subject: Re: C/C++ conversion to IDL
Posted by [James Kuyper](#) on Tue, 07 Jun 2005 16:05:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Michael Wallace wrote:

>> Would there be any preference for C or C++, as i am not really that
>> familiar with either and don't really have the time to learn one and
>> realise later that the other one is more appropriate?
>
> C++ is almost, but not quite, a superset of C. Because of this, I'd
> advise learning C first and then go on to learn C++ should you need to.
> The primary difference between the languages is that C++ supports
> object-oriented programming while C is procedural. If you won't be
> using objects, don't worry about learning C++ now. There are other
> differences between the languages, but none of those make a compelling
> case to learn C++ in this situation.

Most well-written C programs (for a particular definition of "well-written") can be compiled as C++ programs without a change in meaning. So in that sense C++ is "almost" a superset of C. However, there's a great many features of C that are supported by C++, but shouldn't be used when writing C++ code; C++ has much better ways of doing the same thing. Learning how those features work is necessary for an expert C++ programmer, but a bad idea for the novice; the novice might make the mistake of actually using them. Therefore, I'd recommend learning C++ first; learn C only if and when you need it.
