
Subject: Re: From C to IDL questions

Posted by [ryanselk](#) on Fri, 25 May 2007 14:03:04 GMT

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

On May 24, 5:29 pm, b...@hotmail.com wrote:

> On May 23, 3:47 pm, ryans...@gmail.com wrote:

>

>> Ive bought ron klings book, ive read it and done the examples.

>

> You should have all the answers then..

>

>

>

>> My question is this,

>

>> If I have a completed C program, that lets say, has a string output..

>> without adding any additional lines to the C program, can it be called

>> in IDL? Ive done this with SPAWN (thanks to your guys help).. but

>> spawn is not so great (slow, no cross-platform etc).

>

>> It seems ron klings method, some code (or, a fair amount) needs to be

>> added to the C files. Is this just something that im going to have to

>> live with? I ask because I have to convert several files to IDL and im

>> looking to do it as easily as possible.

>

>> Would call_externals be best for this? For various reasons, I cant use

>> call external currently (compiler not reconized and I dont have the

>> discs to re-install right now... "doesnt reconize CL")

>

>> Thoughts or Ideas? Thanks!

>

> If your compiler doesn't work, and your code isn't compatible with

> call_external, and SPAWN is too slow for you, then you are probably

> out of luck. I would suggest the following:

>

> -If your program works as a stand-alone program, then calling it with

> SPAWN is probably the easiest solution.

>

> -If your program is in the form of a DLL, then call_external is the

> way to go. (If you use the proper calling convention, then you may not

> even need to modify the code much.) There are even some automated

> tools for loading functions from an existing DLL without any C

> modifications, although this only works for certain situations.

>

> -Otherwise, wrapping your C functions into a dynamic library and using

> call_external isn't that hard using the online documentation. You

> basically have to make the exported functions use the proper calling

> convention, make sure you cast the input and output variables

> properly, and make sure you don't screw anything up with the memory
> management.
>
> The extra step to converting your code to a DLM isn't that hard, but
> is only justified if you need access to internal IDL functions within
> your C code and you want your program to appear as a built-in IDL
> command.
>
> Brad

Thank you! thats more or less the lines I was thinking.

I was also considering just making some new (full) functions in
another .c
program that just call what I need.. so the C code can do all the work
and ill
just call what I need.

I suppose I dont have to use DLMS, but in the future I may be using an
IDL made
GUI I think this will make things easier.
