

---

Subject: Re: IDL -> C : How to save data from IDL?

Posted by [hotplainrice@gmail.co](mailto:hotplainrice@gmail.co) on Thu, 11 Sep 2008 09:49:43 GMT

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

---

On Sep 10, 9:23 pm, Joost Aan de Brugh <[joost...@gmail.com](mailto:joost...@gmail.com)> wrote:

> On Sep 10, 12:41 pm, "hotplainr...@gmail.com" <[hotplainr...@gmail.com](mailto:hotplainr...@gmail.com)>  
> wrote:

>

>> Hi guys,

>

>> I have managed to get CUDA interfacing to IDL and so far its all good.

>> However, whenever I get problems in C or CUDA, I can't run a debugger

>> because it doesn't recognise IDL. So I want to separate the C code

>> from IDL and this means I have to be able to load data from within C.

>

>> How do I save data that is IDL for C to use?

>

>> I'm thinking of

>

>> running IDL program -> C program which writes the data into a file

>

>> Thanks

>> Zaki

>

> Hello Zaki,

>

> If you use a file:

> Maybe it is the best to use conventional file formats like NetCDF and

> HDF. They are the most portable.

>

> You can probably also use Call\_External (see IDL Help on

> Call\_External).

> I used it for Fortran. IO hope that it works similar for C

> You need a shared object file and a method name.

> The method name is probably something like your defined name plus an

> underscore

> You can see what that name is with "nm <shared object file>" in Linux.

> (I do not know for Windows)

>

> You define a method with arguments argc and argv.

> In Fortran, argv is a list of integers which are the memory addresses

> of your arguments (void-pointers). You should be able to write data

> into these addresses.

> Probably, this only works better with C than with Fortran.

> But if you use this, watch out, because you should always write the

> same size as allocated by IDL. Note that non-long integers in IDL are

> only 2 bytes long and if you memcpy a 4-byte C int there, you write 2

> bytes into unknown territory, ruining other data or causing a

> segmentation fault. (Of course, the former is worse). The other way  
> round is also dangerous. For example, if you write a 4-byte C-float  
> into an address where you have a 8-byte IDL Double, you will get  
> idiotic values in IDL.  
>  
> Best regards,  
> Joost Aan de Brugh

Thanks Joost for the reply, I'll look into it soon.

---