
Subject: Re: variable types in external C modules
Posted by [Nigel Wade](#) on Tue, 31 Oct 2000 10:35:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

"D. Mattes" wrote:

>
> hello IDL gurus:
>
> i am using call_external to run computations in external C modules. i
> want to send arrays of different types to the C module and have them be
> correctly declared in the C module. for example, if the IDL procedure
> sends floats, then the C module needs to declare the incoming arguments as
> floats, and be able to create some scratch memory of type float.
> however, if the IDL procedure sends doubles, then the C module needs to
> declare the variables as doubles. how can i determine the variable types
> of the input arguments from within the C module?
>
> thanks!
> david

I'm fairly sure that the CALL_EXTERNAL interface doesn't pass on any information about the type of variable used in the call. It's pretty much a case of you have to pass what the external code is expecting.

Can you pass an additional argument, the typecode of the array obtained from the size function?

--

Nigel Wade, System Administrator, Space Plasma Physics Group,
University of Leicester, Leicester, LE1 7RH, UK
E-mail : nmw@ion.le.ac.uk
Phone : +44 (0)116 2523568, Fax : +44 (0)116 2523555

Subject: Re: variable types in external C modules
Posted by [Craig Markwardt](#) on Tue, 31 Oct 2000 15:54:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

"D. Mattes" <dmattes@u.washington.edu> writes:

> hello IDL gurus:
>
> i am using call_external to run computations in external C modules. i
> want to send arrays of different types to the C module and have them be
> correctly declared in the C module. for example, if the IDL procedure
> sends floats, then the C module needs to declare the incoming arguments as
> floats, and be able to create some scratch memory of type float.

- > however, if the IDL procedure sends doubles, then the C module needs to
- > declare the variables as doubles. how can i determine the variable types
- > of the input arguments from within the C module?

I think it is common practice to write a "wrapper" routine in IDL. The wrapper routine would perform all of the common error checking, and cast variables to the correct data type. Then the wrapper would call the external C function.

Extensive error checking can be performed, but I believe you need to use the more sophisticated LINK_IMAGE mechanism. In that case all data is passed with their data types intact. The IDL environment does provide some facilities for error checking (see the External Development Guide), but it is tedious. As it turns out, the error checking is most easily performed in the IDL language, ie, in the wrapper.

Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
