Subject: Re: LINKIMAGE problem (beginner)
Posted by davidf on Sat, 07 Mar 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Reinhold Schaaf (Kakadu42@gmx.net) writes:

"LINKIMAGE problem (beginner)". Sigh...

> I want an IDL program call C subroutines. Since the CALL_EXTERNAL
> mechanism seems to provide no possiblities for typechecking (?) I want
> to use LINKIMAGE, where typechecking can (and must) be done. Everything
> is pretty nasty (I really don't like pointers on pointers on pointers
> ...), but anyway.

Here is a piece of free, unsolicited advice:

Do your type checking of variables in IDL *before* you call
your C subroutine with Call_External. I can almost guarantee
that you will be *much* happier with your progress. :-)

Cheers,

David

P.S. You can use the SIZE function for type checking.

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: LINKIMAGE problem (beginner)
Posted by Nigel Wade on Mon, 09 Mar 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Reinhold Schaaf <Kakadu42@gmx.net> writes:

: ----------352DBFD156070D302A7B8B9C
: Content-Type: text/plain; charset=us-ascii
: Content-Transfer-Encoding: 7bit
:
: Hi there,
:
: I want an IDL program call C subroutines. Since the CALL_EXTERNAL
: mechanism seems to provide no possiblities for typechecking (?) I want

: to use LINKIMAGE, where typechecking can (and must) be done. Everything
: is pretty nasty (I really don't like pointers on pointers on pointers
: ...), but anyway.
:
: What I could not find out by myself is:
:
: How can I create (in my C program) a (IDL-) scalar variable of any type,
: that is set to something and returned to IDL. I tried something like:
:
: #include "export.h"
: #include "Linktest.h"
:
: IDL_VPTR margins(int argc, IDL_VPTR argv[])
: {
:
:   IDL_VPTR pvResult = IDL_Gettmp();
:   pvResult = IDL_CvtDbl(1, &pvResult);
:   return(pvResult);
:
: }
:
:
: which should (in my understanding) create a temporary scalar of type
: undefined, convert it to double, and return it back to IDL.
:
: However, after linking everything in with
:
: LINKIMAGE, 'MARGINS', 'Linktest.dll', 1, 'margins', $
:   MAX_ARGS=0, MIN_ARGS=0
:
: (I am working under NT)
:
: and calling the function with
:
: x = margins()
:
: IDL responds:
:
: % MARGINS: Variable is undefined: <UNDEFINED>.
: % Execution halted at: $MAIN$
:
:
: I tried some variants of the above, none worked.
:
: Any help?
:
:
: Thanks

:
: Reinhold
:

What your program does is to first allocate a variable (IDL_Gettmp) and
then attempt to convert that variable to double (IDL_CvtDbl). Is that
what you really intended?

There are two problems with the code as it stands. First, you attempt to
convert a variable which is undefined (the temporary variable returned by
IDL_Gettmp is undefined). Second, you leave the temporary variable dangling
when you assign the converted value to the variable result.
IDL will tidy this up when the routine returns, but it is not good practice.

The reason that IDL is reporting the value of MARGINS as UNDEFINED is that
the value of the result which you return has no type, it is simply converted
from the undefined result of IDL_Gettmp.

If you want to make this simple example work then you can change the code
to something along the lines of:

```
#include "export.h"

IDL_VPTR margins(int argc, IDL_VPTR argv[]) {

  IDL_VPTR dummy = IDL_Gettmp();
  IDL_VPTR result;

  dummy->type = IDL_TYP_LONG;
  dummy->value.l = 123;

  result = IDL_CvtDbl(1,&dummy);
  IDL_Deltmp(dummy);

  return(result);
}
```

or, to convert an input argument

```
#include "export.h"

IDL_VPTR margins(int argc, IDL_VPTR argv[]) {

  IDL_VPTR result;

  if ( argc == 1 )
    result = IDL_CvtDbl(argc, argv);
  else {
```

```
    result = IDL_Gettmp();
    result->type = IDL_TYP_DOUBLE;
    result->value.d = -1.0;
  }

  return(result);
}
```

LINKIMAGE is a flexible method for interfacing to external code,
but you have to be sure you know what you are doing as it is very
easy to crash IDL.

--

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: LINKIMAGE problem (beginner)
Posted by rivers on Thu, 12 Mar 1998 08:00:00 GMT
View Forum Message <> Reply to Message

In article <35016939.EFC1ED89@gmx.net>, Reinhold Schaaf <Kakadu42@gmx.net> writes:
>
> ----------352DBFD156070D302A7B8B9C
> Content-Type: text/plain; charset=us-ascii
> Content-Transfer-Encoding: 7bit
>
> Hi there,
>
> I want an IDL program call C subroutines. Since the CALL_EXTERNAL
> mechanism seems to provide no possiblities for typechecking (?) I want
> to use LINKIMAGE, where typechecking can (and must) be done. Everything
> is pretty nasty (I really don't like pointers on pointers on pointers
> ...), but anyway.

While it is true that CALL_EXTERNAL does not provide typechecking, I don't
think this is really a problem. Every function which you call via
CALL_EXTERNAL should be called through an IDL "wrapper" routine. That wrapper
routine, and ONLY THAT WRAPPER ROUTINE, is what does the CALL_EXTERNAL to your
C code. The wrapper routine converts all arguments passed to it to the correct
data type for the C function. Example:

```
pro my_test, input, output
```

```
; Assume your C code wants its input as a 32 bit int and returns a single
; precision float.  The following code will ensure that input is converted to
; long (if required) and that the output is a float.

output = 0.0
status = call_external('my_lib', 'my_func', long(input), output)
end
```


Mark Rivers (773) 702-2279 (office)
CARS (773) 702-9951 (secretary)
Univ. of Chicago (773) 702-5454 (FAX)
5640 S. Ellis Ave. (708) 922-0499 (home)
Chicago, IL 60637 rivers@cars.uchicago.edu (e-mail)

or:
Argonne National Laboratory (630) 252-0422 (office)
Building 434A (630) 252-0405 (lab)
9700 South Cass Avenue (630) 252-1713 (beamline)
Argonne, IL 60439 (630) 252-0443 (FAX)