Subject: Re: External c calls broken in IDL 5.5
Posted by Bob[1] on Wed, 04 Sep 2002 21:01:42 GMT
View Forum Message <> Reply to Message

Stein Vidar Hagfors Haugan wrote:

> Now, for the crashing, I bet your fortran routine uses complex
> numbers?? (Okay, I don't bet a lot, but this is my only "known"
> suspect). Guess what - IDL changed the spec of the IDL_CvtComplex
> call in IDL 5.4 and told no-one about it (in the edg.pdf at least, at
> the time). So, I suspect this is your problem (I haven't tested
> dlmform on 5.5, since I don't run that here.. be my guest).

No, it was totally my fault (see my other message).

> I know that there are a few anachronisms that should be updated in the
> code for dlmform... Supporting the new types like 64 bit types is one,
> but unless I see or hear of a need for it (I'm assuming this is mostly
> used for legacy code in fortran, and it wouldn't have those as inputs,
> would it?). Here's a deal: If you send me a list of the obsolete
> functions that you'd like to see replaced (with the replacements and,
> preferably, the IDL version in which the change was made!!), I'll see
> what I can do!

Well, the calls to IDL_AddSystemRoutine in IDL_Load is obsolete and should be replaced with
IDL_SysRtnAdd which uses IDL_SYSFUN_DEF2 structure instead of the IDL_SYSFUN_DEF structure (I think this changed in IDL 5.4 but am not sure). However, it does still work the way you have it.

A bigger problem I have is that the array dimensions for output arrays are usually off by one in the c-code generated by dlmform when there is an array in the fortran code. For example the fortran program:

```
subroutine test(x,y,n)
integer n
real x(n), y(n)
do i=1,n
   y(i) = x(i)*x(i)
enddo
end
```

The c-code (test.c) has the following block where the variable y is defined:

```
in = 0;          /* Y_ : REAL : (N) :  */
if (in) {
 } else { /* Output */
  IDL_EXCLUDE_EXPR(Y_); /* Output cannot be expression */
```

```
  ndim = 1;
  dim[0] = N_->value.l+1 /*???*/;
  IDL_StoreScalarZero(Y_,IDL_TYP_FLOAT);  /* Free resources */
   IDL_MakeTempArray(IDL_TYP_FLOAT,ndim,dim,IDL_ARR_INI_ZERO,&a mp;tmp);
  IDL_VarCopy(tmp,Y_);
}
```

The dim[0]=N_->value.l+1 should be dim[0]=n_value.l. I've gotten in the habit of always changing these but it would be nice I didn't have to.

And last, I'll ask you this since you seem to know you way around. Is it possible to catch errors in the fortran code so that one ends up at the IDL prompt instead of core dumping IDL?

Thanks.

Bob

---