Subject: Re: How to traverse/inquire a class object structure in IDL?
Posted by davidf on Wed, 13 Oct 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Paul van Delst (paul.vandelst@ssec.wisc.edu) writes:

> Last night I entered into the world of IDL objects. I was amazed at how much
> easier it is to keep control of a data object rather than using a regular
> structure.

Hooray!

> Anyway, since I have been programming IDL in an Object Oriented mode for about 8
> hours, I have some questions that I hope someone out there can help me with. The
> documentation (on-line and printed) was not useful.

No, probably not. :-(

> I have a class structure definition in nasti__define.pro:.
> [Much deleted.]

```
>  This all works fine. I have an simple inquire method:
>
>  PRO nasti::inquire_nasti
>
>    PRINT, FORMAT = '( /5x, "Inquiring..." )'
>    PRINT, PTR_VALID(), OBJ_VALID()
>
>  END
>
>  which when I run it, gives:
>
>  IDL> n->inquire_nasti
>       Inquiring...
>  <PtrHeapVar2>
>  <ObjHeapVar1(NASTI)>
>
>  where the PtrHeapVar2 is the pointer to "self.wavenumber" and the object
>  reference is for the object. Cool.
```

Cool if you have one object. Not so cool if you have several
other programs with objects running, probably. :-)

I think I would have written it something like this:

```
PRO nasti::inquire
   Print, 'Wave Number: ', *self.wavenumber
   Help, *.self.radiance, Output=thisOutput
```

```
  Print, 'Radiance Represented As: ', thisOutput
END
```

> Not good. As more objects are created and destroyed, the valid pointer list
> grows. I would like to do the following in a CLEANUP method:
>
>   FOR i = 0, n_object_structure_elements - 1 DO $
>   IF (PTR_VALID(self.(i))) THEN $
>   PTR_FREE, self.(i)
>
> that is, *explicitly* free up the pointers. This works great if I have a value
> for n_object_structure_elements.
>
> QUESTIONS:
>
> 1) Is my technique valid? That is, I want to do the following:
> - create a data object
> - read some amount of data into that object
> - do stuff with the data object
> - delete the data object INCLUDING any pointers in the object.
> I don't know how much data I have ahead of time so I used pointers. Can I create
> data objects on the fly, based on how much data is in a datafile or requested
> from a datafile?
>
> 2a) If my technique is o.k., how do I free up the pointers in my object before I
> destroy it?

Your technique is probably OK, but it seems a bit
convoluted to me. Why not just write the CLEANUP routine
like this:

```
  PRO nasti::cleanup
  Ptr_Free, self.wavenumber
  Ptr_Free, self.radiance
  ...
  Ptr_Free, self.decimal_time
  END
```

A few extra keystrokes, perhaps, but it has the advantage
that you can see at a glance what it does. :-)

> ..OR..
>
> 2b) Is the above code stub a valid/smart way to free up the pointers in a data
> object and, if so, how do I determine the value of n_object_structure_elements?
> (You can't use N_TAGS() on an object but you can use the self.(i) type of
> structure reference so I'm confused.)

If you really like your solution, you could find
the number of fields in your object like this:

```
   thisClass = Obj_Class(self)
   ok = Execute("struct = {" + thisClass + "}")
   object_structure_elements = N_Elements(Tag_Names(struct))
```

But this just seems way too clever for me. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155