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:

> First off, thanks to David and Pavel for their insights. Before I could check the
> newsgroup for replies, one of our younger go-getter science types came and told me
> something about object oriented programming that made good sense:
>
> The data should be an attribute of the object, not the object itself.

Yeah, those young bucks know about object programming. :-(

```
> Hmm. Anyway, he and I sat down for about 15 minutes and came up with the following
> class structure definition and cleanup method:
>
> PRO nasti__define
>
> ; -- Define the NAMED data structure attribute
>   data = { data, $
>            wavenumber      : PTR_NEW(), $
>            radiance        : PTR_NEW(), $
>            altitude        : PTR_NEW(), $
>            fov_angle       : PTR_NEW(), $
>            fov_index       : PTR_NEW(), $
>            latitude        : PTR_NEW(), $
>            longitude       : PTR_NEW(), $
>            aircraft_roll   : PTR_NEW(), $
>            aircraft_pitch  : PTR_NEW(), $
>            scan_line_index : PTR_NEW(), $
>            date            : PTR_NEW(), $
>            time            : PTR_NEW(), $
>            decimal_time    : PTR_NEW() }
>
> ; -- Create object CLASS structure
>   nasti = { nasti, $
>             data : data }
>
> END
>
> I like this becuase now I can add additional attributes whenever I want, e.g.
> global attribute data read from the netCDF data file containing instrument
> calibration information and/or processing software CVS/RCS info etc.
```

Not too bad, although I thought the first one was OK too.
Actually, it is the data and the methods that manipulate
the data that should be encapsulated in the object, so I

don't see that this new construction gains us much of anything,
except more structure de-referencing. :-)

But I *would* change the name of the structure from DATA
to something just a tad less generic. I see plenty of
trouble coming down the road with a name like DATA.

```
>  The cleanup method is now:
>
>  PRO nasti::cleanup
>
>    PRINT, FORMAT = '( /5x, "Clean up..." )'
>
>  ; -- Free up pointers
>    n_data_fields = N_TAGS( self.data )
>    FOR i = 0, n_data_fields - 1 DO $
>      IF ( PTR_VALID( self.data.(i) ) ) THEN $
>        PTR_FREE, self.data.(i)
>
>  END
```

Have to admit that this is compact. :-)

> I wish I'd "discovered" objects earlier......all that code I wrote that *needs* the
> data to be encapsulated. Crikey.

You are on a slippery slope here. Once you fall
for objects almost *everything* looks like a perfect
opportunity to use one. :-)

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