Subject: Need advice on building an object container
Posted by Paul Van Delst[1] on Mon, 27 Apr 2009 14:41:53 GMT
View Forum Message <> Reply to Message

Hello,

I need a bit of advice on building a container for object.

I have an object definition, let's say 'RTS', that contains the simulated top-of-atmosphere radiances for various satellites. I also have files that contains all the instance data for a whole bunch of those objects. Those data are organised by the sensor channel (i.e. different frequencies) and atmospheric profiles. So, in my pre-object-code I would create a structure array after determining the dimensions, something like

```
Rts = PTRARR( n_Channels, n_Profiles )
FOR m = 0L, n_Profiles-1L DO BEGIN
  FOR l = 0L, n_Channels-1L DO BEGIN
    Rts[l,m] = PTR_NEW({CRTM_RTSolution})
    ...read the current record...
  ENDFOR
ENDFOR
```

But now I want to use objects (mostly as a learning exercise, but also to stop folks from mucking about with the innards of the structure when they use it).

So, the structure definition of RTS doubles as the object defn also. What do people recommend for reading the datafile? I see two options:

1) Use OBJARR(). Read dimensions and create the array and then fill it. However, this would still require the user to create the array after inquiring the file for its dimensions, and basically keep track of things.

2) Use a container. This is what I came up with this morning:

```
PRO RTSfile__Define
  void = { RTSfile, $
           Filename : '', $
           FileId   : 0L, $
           INHERITS IDL_Container }
END
```

and, in the RTSfile::Read method I would simply read each RTS object from the file and do a

```
rtsfile->Add, rts
```

But this method doesn't preserve the basic [n_Channels, n_Profiles] structure of the data. And, there's no indication in the RTSfile definition that this is a container for RTS objects, rather than a generic container -- but I'm wondering if worrying about that is just a distraction?

3) Sort-of combining (1) and (2) doing something like:

```
  PRO RTSfile__Define
    void = { RTSfile, $
             Filename : '', $
             FileId   : 0L, $
             n_Channels : 0L, $
             n_Profiles : 0L, $
             rts : PTR_NEW() }
  END
```
where, eventually,
```
  rts = PTR_NEW(OBJARR(n_Channels, n_Profiles))
```
and then fill in the object references to the RTS object.

So I was wondering how people would structure their code to handle this sort of thing. I think it's a common enough paradigm that a pattern probably exists but I haven't found it.

Any info, hints, tips, appreciated.

cheers,

paulv

Subject: Re: Need advice on building an object container
Posted by Mike[2] on Thu, 30 Apr 2009 18:00:15 GMT
View Forum Message <> Reply to Message

Have you looked at the IDL_Container object? It could be inherited by your containers and used to handle all the bookkeeping.

Mike

Subject: Re: Need advice on building an object container
Posted by Paul Van Delst[1] on Fri, 01 May 2009 14:26:46 GMT
View Forum Message <> Reply to Message

Mike wrote:
> Have you looked at the IDL_Container object? It could be inherited by
> your containers and used to handle all the bookkeeping.

Umm... yes. That's why I used it in my original example:

```
>   PRO RTSfile__Define
>     void = { RTSfile, $
>              Filename : '', $
>              FileId   : 0L, $
>              INHERITS IDL_Container }
```

> END

:o)

My problem is with the "packaging" of various levels of these objects. Or, alternatively, translating my real world organisation of the data into something amenable to OOP programming. My goal here is somewhat an academic exercise, but I want to be able to easily change the definition of my base object without having to retool a bunch of code.

I think Ben's advice is what I needed. Just a pattern on which I can work with, or build on.

cheers,

paulv
