Subject: Re: Structure assign question
Posted by btt on Mon, 26 Nov 2001 17:20:38 GMT
View Forum Message <> Reply to Message

Hi Paul,

One approach you might consider is making each of your PDprofiles data structures into objects and
your PDdata structure an IDL container. I find this approach easier than wrestling with arrays of pointers to structures with pointers that have other pointers... well, you know what I mean.
Also, you don't have to know how many profiles must be stored ahead of time. I prefer to base all of my structures on Martin Schultz's MGS_BaseObject and I only use his MGS_Container in which he
has added a great many useful features to the basic IDL_Container class.

```
PRO PDdata__define

   struct = {PDData, $
   INHERITS MGS_Container}

END

; -- Loop over profiles
  FOR m = 0L, n_profiles - 1L DO BEGIN

   READU, file_id, k, j

   profile = PDprofile_create( k, j )    ;<--- this would have to return an object
                                         ;<---- with all the bells and whistles
                                         ;<-----like Init, GetProperty, SetProperty
and Cleanup
   self->Add,  profile

   n_layers[ m ]    = k
   n_absorbers[ m ] = j

  ENDFOR
```

Ben

> Anyway, I have the following object/structure definition procedure:
>
```
>  PRO PDdata__define
>    PDcommon_definition
>    COMMON PDcommon
>    data_structure = { PDdata, n_profiles : IO_INT_TYPE, $
```

> profile : PTR_NEW() }
> END ; PRO PDdata__define
>
> nice and simple.
>
> In my PDdata::Read method, which fills the above data structure/object, I have the following:
>
> ; -- Create the top level PDdata structure
> self.n_profiles = n_profiles
> self.profile = PTR_NEW(REPLICATE({ PDprofile }, n_profiles))
>
> ; ----------
> ; Each profile dimension
> ; ----------
>
> ; -- Initialise arrays
> n_layers = LONARR(n_profiles) & k = 0L
> n_absorbers = LONARR(n_profiles) & j = 0L
>
> ; -- Loop over profiles
> FOR m = 0L, n_profiles - 1L DO BEGIN
>
> READU, file_id, k, j
>
> profile = PDprofile_create(k, j)
> (*self.profile)[m] = profile
>
> n_layers[m] = k
> n_absorbers[m] = j
>
> ENDFOR
>
> My question relates to the PDprofile_create() function. It returns a named structure (name is
> "PDprofile"). This structure is replete with various other (named) structures and pointers etc.
> hence the separate function. I then assign this structure to the required element of the data
> object, self.profile :
>
> (*self.profile)[m] = profile
>
> Here is where the question may or may not be considered brain-dead. Is this sort of (named)
> structure assign o.k., i.e. does the entire heirarchy of pointers and structures within the
> "profile" structure variable get assigned (it appears so at this point but I'm still testing).
> And am I creating a separate instance of the PDprofile structure or am I confusing the pointer
> with what the pointer is pointing at (get it?).
>

--
Ben Tupper

Bigelow Laboratory for Ocean Science
180 McKown Point Road
West Boothbay Harbor, ME 04575
www.bigelow.org
btupper@bigelow.org