Subject: Re: IDL objects (not object graphics) tutorial?
Posted by Paul Van Delst[1] on Fri, 02 Dec 2005 16:30:07 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Paul Van Delst writes:
>
>
>> Your webpage is great! The one new thing I learned:
>>
>>   self.ptr=ptr_new(/allocate)
>>
>> can be used to subsequently point to anything without further allocation!? E.g. from your
>> tutorial:
>>
>>   a->set,image
>> or
>>   a->set,!d
>>
>> where in the set method, the value is simply assigned:
>>   *(self.ptr)=value
>>
>> Excuse my brain-deadedness, but how is this possible? I looked at the IDL docs but there
>> is (surprise, surprise) no elaboration about this little nugget of information regarding
>> PTR_NEW. Wouldn't susbsequent calls like the above cause a memory leak, e.g.
>>
>> IDL> image=findgen(512,512)
>> IDL> a=obj_new('data')   ;-- create object variable a
>> IDL> a->set,image   ;-- insert image
>> IDL> a->set,!d
>>
>> What would happen to the "image" data?
>
>
> The deep answers are all contained in the Pointer Tutorial:
>
>   http://www.dfanning.com/misc_tips/pointers.html

Ahhh! It's just like a regular IDL variable. D'oh! - now why didn't assume that to be the
case in the first place?!

Very cool. I was treating this sort of thing like I do in Fortran95 where the pointers are
typed (i.e. they can only point to certain data types, kinds, and ranks) and that leads to
a lot of code bloat.

This changes everything. <monty>Excellent</monty> :o)

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC