Subject: Re: pointers in IDL
Posted by Thomas A. McGlynn on Tue, 29 Apr 1997 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Christian Oehreneder <co@ipf.tuwien.ac.at> writes:
>
>> My application uses a not a priori known number of images with different
>> sizes, which I want to access easily and flexible, preferrably by
>> writing img_i = img(i) or something similar. Because of the different
>> sizes of the images this is not easily done in IDL.
>> In common programming lanuages it would be natural to have an array of
>> pointers to each of the images.
>> So far I have not found a possibilty to do this in IDL. Handles are nice
>> but they do not allow to have two pointers on the same data. The effect
>> is that once you take the value of a handle (without copying it of
>> course !!) the handle itself contains an undefined value. It is not
>> possible to have a list of images and pass them to some manipulation
>> routines by pointer, e.g. multiplying all images by a factor of 2. Of
>> course it is possible to take the value form the handle, multiply it and
>> set it as value of the handle, but that's rather cumbersome.
>>
>> Has anyone an idea how to work arround this??
>
> I think there is no good way to work around the cumbersome
> method of handles except to upgrade to IDL 5.0 when it comes
> out and take advantage of the new POINTER data type. I'm sure
> it was added to the IDL language for exactly this reason.
>
> Cheers!
>
> David
>
> ---

One way to deal with this if the number of images isn't large
and you don't want to discard or rearrange them can be to use structures to
act as containers for your images.

If you know what the images are then you can just define the structure naturally.
If you need to build it incrementally then you can do something like:

```
   mystruct =
      create_struct(mystruct,'e'+string(n_tags(old_struct)),new_im age)
```
[this particular constructions makes it difficult for you to use
 the tagnames, but I'm assuming that's not an issue.]

You have to handle the first one specially of course.

Now you can pass the structure to subroutines and do whatever you want.
To access a given image just:

```
   image_i_want = my_struct.(index)
```

which seems to be close to what you want.   This may only get
you back to IDL 4.0 though, I'm not sure if create_struct works this way
(or at all) much earlier.

Tom McGlynn