Subject: Re: Initializing object array
Posted by Dick Jackson on Tue, 09 Mar 2004 05:45:12 GMT
View Forum Message <> Reply to Message

Hi David,

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.1ab6e521618e56ee9896f1@news.frii.com...
>
> I have an object. One of the fields of this objects
> is a object array. The field is called "contours":
>
> PRO myclass__define
> class = { MYCLASS, contours:Obj_New()}
> END
>
> Now, when I create the object, I want to pass an object
> array of ROI objects that I created somewhere else.
>
> FUNCTION myclass::INIT, Contours=contours
> self.contours = contours
> RETURN, 1
> END
>
> This doesn't work. Says contours must be a scalar
> in this context. Well!!! Shucks.
>
> Surely I have done this before. But I can't for the
> life of me remember how. How do I initialize a field
> as an object array?

  class = { MYCLASS, contours:ObjArr(nElements)}

would do it, but it will be a fixed number of elements, and the passed
'contours' would have to match that. If that's what you need, then fine,
but I bet you need flexibility. All I can see for a solution right now
is using a pointer:

  class = { MYCLASS, contours:Ptr_New(/Allocate_Heap)}

then, to assign it:
  *self.contours = contours

and to refer to one contour:
  (*self.contours)[i]

Sorry if I'm stating the obvious... or am *I* missing something?

Cheers,
--
-Dick

Dick Jackson / dick@d-jackson.com
D-Jackson Software Consulting / http://www.d-jackson.com
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392