
Subject: Re: Constant Size Array in Class Init
Posted by [Chris Williams](#) on Mon, 23 May 2011 21:59:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

On May 23, 5:32 pm, David Fanning <n...@idlcoyote.com> wrote:

> Chris Williams writes:

>> I am creating a class that holds multidimensional data of several
>> different arrays. What I am trying to do is create a class to hold the
>> arrays, and then define the array sizes in the init function since
>> they will be different sizes.

>

>> For example, I want to do something like this

>

>> FUNCTION myclass:init, dim1, dim2, dim3

>

>> myarray = dblarr(dim1, dim2, dim3)

>

>> RETURN,1

>> END

>

>> PRO myclass__define

>

>> void = {myclass, myarray :dblarr } ;NO SIZE ON ARRAY

>

>> RETURN

>> END

>

>> However, I get a conflicting data structure error without first
>> defining the size of the array in the procedure myclass__define. Is
>> there an empty array object that allows me to define the array size at
>> initialization and not at object creation?

>

> You are going to have to define this field as a pointer

> to an array in IDL.

>

> void = {myclass, myarray :ptr_new() } ;NO SIZE ON ARRAY

>

> Then,

>

> myArray = Ptr_New(dblarr(dim1,dim2,dim3))

> Help, *myArray

>

> Cheers,

>

> David

>

> --

> David Fanning, Ph.D.

- > Fanning Software Consulting, Inc.
- > Coyote's Guide to IDL Programming:<http://www.idlcoyote.com/>
- > Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Hmm, now I get "Pointer expression not allowed in this context" when entering the initialization function.
