
Subject: Re: Constant Size Array in Class Init
Posted by [David Fanning](#) on Mon, 23 May 2011 21:32:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

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.")

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.

Subject: Re: Constant Size Array in Class Init
Posted by [Chris Williams](#) on Mon, 23 May 2011 22:03:47 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.")

I just realized that all the fields of the class are private by default and there is no way I am going to write 50 setters and getters for a data storage class. I'm just going to do what I need to do in python.

Thanks a ton David.

Subject: Re: Constant Size Array in Class Init
Posted by [penteado](#) on Mon, 23 May 2011 22:11:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

On May 23, 7:03 pm, Chris Williams <fam36boozl...@gmail.com> wrote:
> I just realized that all the fields of the class are private by
> default and there is no way I am going to write 50 setters and getters
> for a data storage class. I'm just going to do what I need to do in
> python.

Just a couple of ideas:

It does not have to be 100 functions. It can be just two (get/setproperty), either each one with 50 arguments, or just one argument, with uses of execute() or scope_varfetch().

But in this case maybe just putting an extra layer would be more convenient. You can group everything into a single container (by a pointer, a structure, a hash or a list, whichever is more convenient), and then pass just that one container with get/setproperty.

Subject: Re: Constant Size Array in Class Init
Posted by [David Fanning](#) on Mon, 23 May 2011 22:16:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

Chris Williams writes:

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

Did you restart IDL after you changed the class definition?

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.")

Subject: Re: Constant Size Array in Class Init
Posted by [David Fanning](#) on Mon, 23 May 2011 22:18:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

Chris Williams writes:

- > I just realized that all the fields of the class are private by
- > default and there is no way I am going to write 50 setters and getters
- > for a data storage class. I'm just going to do what I need to do in
- > python.

Goodness, I wouldn't write 50 getters and setters either!!
Sounds like JAVA to me. :-)

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.")
