
Subject: Re: Q: pointers inside of an array of structures?
Posted by [Ivan Zimine](#) on Wed, 16 Aug 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Amara Graps wrote:

```
>  
> IDL> thisstruc = {orbit:"", freq:ptr_new()}  
> IDL> periodcube = replicate(thisstruc,1)  
>  
> ;Put values into the structure  
> IDL> periodcube(0).orbit='G2'  
> IDL> *periodcube(0).freq=DINDGEN(50)  
> % Unable to dereference NULL pointer: <POINTER (<NullPointer>)>.  
> % Execution halted at: $MAIN$
```

x = ptr_new() doesn't create a new heap variable so you can't assign anything to *x

you should try with:

```
IDL> thisstruc = {orbit:"", freq:ptr_new(/alloc)}
```

or with some value

```
IDL> thisstruc = {orbit:"", freq:ptr_new(!values.f_nan)}
```

Ivan

--

Ivan Zimine | ivan.zimine@physics.unige.ch
Dpt. of Radiology | (+41 22) 372 70 70
Geneva University Hospitals |

Subject: Re: Q: pointers inside of an array of structures?
Posted by [davidf](#) on Wed, 16 Aug 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Amara Graps (Amara.Graps@mpi-hd.removethis.mpg.de) writes:

```
> But what about varying data lengths inside of the array of  
> structures, for example, if my freq arrays are each a different  
> length:  
>  
> IDL> thisstruc = {orbit:"", freq:ptr_new()}  
> IDL> periodcube = replicate(thisstruc,1)  
>  
> ;Put values into the structure
```

```
> IDL> periodcube(0).orbit='G2'  
> IDL> *periodcube(0).freq=DINDGEN(50)  
> % Unable to dereference NULL pointer: <POINTER (<NullPointer>)>.  
> % Execution halted at: $MAIN$  
>  
> What am I doing wrong?
```

Uh, trying to dereference a NULL pointer, it looks like. :-)

You can create a NULL pointer like this:

```
IDL> ptr = Ptr_New()
```

You can tell this is a NULL pointer because it is invalid:

```
IDL> Print, Ptr_Valid(ptr)  
0
```

And, of course, it is illegal to dereference an invalid pointer:

```
IDL> *ptr = 5  
% Unable to dereference NULL pointer: <POINTER (<NullPointer>)>.
```

What I think you want in this case is a pointer to an undefined variable. Another way to say this is that you want to allocate some memory on the heap, but you don't want that memory to actually point to anything yet. One way to do that is like this:

```
IDL> goodPtr = Ptr_New(/Allocate_Heap)
```

Now, even though this doesn't point to anything, it *is* a valid pointer:

```
IDL> Print, Ptr_Valid(goodPtr)  
1
```

This means it can be dereferenced:

```
IDL> *goodPtr = 5
```

The only caveat is that you have to be just as careful using this dereferenced pointer in an expression as you do using an undefined variable.

So, you have a couple of choices in your program. One choice (this one immediately solves your problem) would be creating your initial structure like this:

```
thisstruc = {orbit:", freq:ptr_new(/allocate_heap)}  
periodcube = replicate(thisstruc,1)  
*periodcube(0).freq=DINDGEN(50)
```

The other choice would be to check if you have a valid pointer before you dereference it:

```
thisstruc = {orbit:", freq:ptr_new()}  
periodcube = replicate(thisstruc,1)  
IF Ptr_Valid(periodcube(0).freq) THEN *periodcube(0).freq=DINDGEN(50) $  
ELSE periodcube(0).freq = Ptr_New(DINDGEN(50))
```

Now you are free to put whatever you like in each structure's pointer. In other words, the frequency field can not be just variable length arrays, it can be completely different data!

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Q: pointers inside of an array of structures?
Posted by [Theo Brauers](#) on Wed, 16 Aug 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Amara:

I think you are looking for the following:

```
IDL> thisstruc = {orbit:", freq:ptr_new()}  
  
IDL> periodcube = replicate(thisstruc,1)  
IDL> periodcube[0].orbit='G2'  
IDL> periodcube[0].freq=PTR_NEW(DINDGEN(50))  
  
IDL> temppperiod = thisstruc  
IDL> temppperiod.orbit='G3'  
IDL> temppperiod.freq=PTR_NEW(DINDGEN(75))  
  
IDL> newperiodcube=[periodcube,temppperiod]
```

```
IDL> HELP, newperiodcube[0], /STR
IDL> HELP, *newperiodcube[0].freq
IDL> HELP, newperiodcube[1], /STR
IDL> HELP, *newperiodcube[1].freq
```

which outputs:

```
** Structure <13585f8>, 2 tags, length=12, refs=5:
  ORBIT      STRING  'G2'
  FREQ       POINTER  <PtrHeapVar10>
<PtrHeapVar10> DOUBLE  = Array[50]
** Structure <13585f8>, 2 tags, length=12, refs=5:
  ORBIT      STRING  'G3'
  FREQ       POINTER  <PtrHeapVar11>
<PtrHeapVar11> DOUBLE  = Array[75]
```

Cheers Theo

Amara Graps wrote:

```
>
> Hi Folks,
>
> I am trying to create and build a flexible data structure
> in IDL for a situation where we have fields of different types
> in a structure, and then that structure is made into an array
> of structures using "replicate". And I extend the
> array of structures by concatenating, because I don't
> know how many structures I need.
>
> My question: Can a field inside of the array-of-structures be
> a pointer ?
>
> For example --
>
> I know how to flexibly extend a structure,
> with each of the fields defined the same way:
>
> -----
> Create an array of an anonymous structure
> IDL> thisstruc = {orbit:", len:0,freq:DBLARR(100)}
> IDL> periodcube = replicate(thisstruc,1)
>
> ;Put values into the structure
> IDL> periodcube(0).orbit='G2'
> IDL> periodcube(0).len=50 ;to store length of array : freq
> IDL> periodcube(0).freq=DINDGEN(50)
>
> ;Create a temporary structure like the original
```

```

> IDL> tempperiod = thisstruc
>
> ;Put values into the temporary structure
> IDL> tempperiod.orbit='G3'
> IDL> tempperiod.len=75
> IDL> tempperiod.freq=DINDGEN(75)
>
> ;Now extend it by concatenating
> IDL> newperiodcube=[periodcube,tempperiod]
>
> ;Take a look
> IDL> help, newperiodcube, /st
> ** Structure <89af0>, 3 tags, length=816, refs=4:
>   ORBIT      STRING   'G2'
>   LEN        INT      50
>   FREQ       DOUBLE   Array[100]
> IDL> print, newperiodcube(0).orbit, ' ', newperiodcube(1).orbit
> G2 G3
> IDL> print, newperiodcube(0).len, ' ', newperiodcube(1).len
>   50    75
> -----
>
> But what about varying data lengths inside of the array of
> structures, for example, if my freq arrays are each a different
> length:
>
> IDL> thisstruc = {orbit:"", freq:ptr_new()}
> IDL> periodcube = replicate(thisstruc,1)
>
> ;Put values into the structure
> IDL> periodcube(0).orbit='G2'
> IDL> *periodcube(0).freq=DINDGEN(50)
> % Unable to dereference NULL pointer: <POINTER (<NullPointer>)>.
> % Execution halted at: $MAIN$
>
> What am I doing wrong?
>
> Is it possible to have pointers inside of an array of structures?
>
> My pointer programming experiences were 15 yrs ago in a Pascal
> class.. it's quite possible I'm missing something big, with how
> the pointers work...
>
> Thanks very much in advance,
>
> Amara
>
> P.S. IDL Version 5.3 (sunos sparc)

```

```
> --
>
> *****
> Amara Graps          | Max-Planck-Institut fuer Kernphysik
> Interplanetary Dust Group | Saupfercheckweg 1
> +49-6221-516-543      | 69117 Heidelberg, GERMANY
>                        * http://galileo.mpi-hd.mpg.de/~graps
> *****
> "Never fight an inanimate object." - P. J. O'Rourke
```

--

Theo Brauers
Institut fuer Atmosphaerische Chemie (ICG-3)
Forschungszentrum Juelich
52425 JUELICH, Germany
Tel. +49-2461-61-6646 Fax. +49-2461-61-5346
<http://www.kfa-juelich.de/icg/icg3/MITARBEITER/th.brauers.ht ml>
