Subject: Structure Creation with Execute
Posted by msreeve on Tue, 11 Apr 1995 07:00:00 GMT
View Forum Message <> Reply to Message

I am reading in an HDF file containing 30-40 Scientific Data Sets of
varying size and rank. The sizes are not known ahead of time.
I would like to read each data set into an element of an anonymous
structure and output that from the function. Currently I construct
a structure definition string by going through the HDF file and
collecting the dimensions of each SDS. I then execute it, to define
the structure (with tag names HDF0, HDF1, ...), and then make another
pass through the file to read the data into the structure.

This works well if there are only a few data sets in the file, but
if the structure definition string exceeds 131 characters (IDL's
ridiculously-low maximum for an EXECUTE), the EXECUTE fails.
Is there a way around this somehow? Is it possible to append one
structure to another (I thought not)? Any suggestions?

Mark Reeve, Ph.D.
msreeve@netcom.com OR mreeve@xti.com

Subject: Re: Structure Creation with Execute
Posted by wmc on Tue, 18 Apr 1995 07:00:00 GMT
View Forum Message <> Reply to Message

In article GLB@netcom.com, msreeve@netcom.com (Mark D. Reeve) writes:
>
> This works well if there are only a few data sets in the file, but
> if the structure definition string exceeds 131 characters (IDL's
> ridiculously-low maximum for an EXECUTE), the EXECUTE fails.
> Is there a way around this somehow? Is it possible to append one
> structure to another (I thought not)? Any suggestions?

The best way around this is to use "create_structure" ;-)

I came across the same problem (actually when porting pv-wave code to idl, when
an unannounced feature of a new idl release was to cut down the max number of
characters in an execute string from 256 to the 131 that it says in the manual!)
and the answer is, don't use execute, use create_structure (does appends too).

- William

Subject: Re: Structure Creation with Execute
Posted by Greening C M on Wed, 19 Apr 1995 07:00:00 GMT

Hi,

If I understand correctly what you are trying to do you may be
able to nest structures within structures to create a sort of
linked list. If you need to make modifications to list (inserting,
or deleting items then it get a bit more complicated.

Chris
greec@essex.ac.uk

```
; place a new structure onto the front of the list
FUNCTION My_struct, x, r
  RETURN, {, data:x, rest:r}
END

; test the list
PRO Test
  list = 0

  ; set up the list to contain a set of different length arrays
  FOR i = 1, 11 DO BEGIN
    ; create some junk data to store in the list
    j = indgen(i)
    ; add the data to the list
    list = my_struct(j, list)
  ENDFOR

  ; print the contents of the list
  tmp = list
  FOR i = 0, 10 DO BEGIN
    print, tmp.data
    tmp = tmp.rest
  ENDFOR
END
```

Subject: Re: Structure creation
Posted by Carsten Lechte on Wed, 20 Jun 2012 10:21:42 GMT

On 20/06/12 12:07, John Coxon wrote:
> Hi there,
>
> Having an odd issue with creating structures.
>
> idl > med = {avg1, dawnR1:0.0, duskR1:0.0, dawnR2:0.0, duskR2:0.0}

> idl > print,med.dawnR1
> 0
> idl > help,med.dawnR1
> <Expression> INT = 0

I would guess that you previously defined a structure with name avg1
that had dawnR1 as an integer. dawnR1's type is determined by the
initialisiation value from the very first use of the structure. A
naked "40" or "0" is interpreted by IDL as an integer constant.

Either change to using unnamed structures, or be extra careful when
specifying values, e.g. always do 1.0 when you want float, 1d0 when
you want double, 40L for long etc.

chl

Subject: Re: Structure creation
Posted by John Coxon on Wed, 20 Jun 2012 11:01:15 GMT
View Forum Message <> Reply to Message

On 20/06/2012 11:21, Carsten Lechte wrote:
> On 20/06/12 12:07, John Coxon wrote:
>> Hi there,
>>
>> Having an odd issue with creating structures.
>>
>> idl > med = {avg1, dawnR1:0.0, duskR1:0.0, dawnR2:0.0, duskR2:0.0}
>> idl > print,med.dawnR1
>> 0
>> idl > help,med.dawnR1
>> <Expression> INT = 0
>
> I would guess that you previously defined a structure with name avg1
> that had dawnR1 as an integer. dawnR1's type is determined by the
> initialisiation value from the very first use of the structure. A naked
> "40" or "0" is interpreted by IDL as an integer constant.
>
> Either change to using unnamed structures, or be extra careful when
> specifying values, e.g. always do 1.0 when you want float, 1d0 when you
> want double, 40L for long etc.

I see, that would make sense. Is there any way to clear a structure, for
instance, between two runs of a program?

--
John Coxon

Subject: Re: Structure creation
Posted by lecacheux.alain on Wed, 20 Jun 2012 14:06:06 GMT
View Forum Message <> Reply to Message

On 20 juin, 13:01, John Coxon <john.co...@gmail.com> wrote:
> On 20/06/2012 11:21, Carsten Lechte wrote:
>
>
>
>
>
>> On 20/06/12 12:07, John Coxon wrote:
>>> Hi there,
>
>>> Having an odd issue with creating structures.
>
>>> idl > med = {avg1, dawnR1:0.0, duskR1:0.0, dawnR2:0.0, duskR2:0.0}
>>> idl > print,med.dawnR1
>>> 0
>>> idl > help,med.dawnR1
>>> <Expression> INT = 0
>
>> I would guess that you previously defined a structure with name avg1
>> that had dawnR1 as an integer. dawnR1's type is determined by the
>> initialisiation value from the very first use of the structure. A naked
>> "40" or "0" is interpreted by IDL as an integer constant.
>
>> Either change to using unnamed structures, or be extra careful when
>> specifying values, e.g. always do 1.0 when you want float, 1d0 when you
>> want double, 40L for long etc.
>
> I see, that would make sense. Is there any way to clear a structure, for
> instance, between two runs of a program?
>
> --
> John Coxonhttp://www.chickensinenvelopes.net/
>
>

If your structure has only numeric fields:

for i=0,N_Tags(your_struct)-1 do your_struct.(i) = 0

alx.

Subject: Re: Structure creation
Posted by Michael Galloy on Wed, 20 Jun 2012 16:11:04 GMT
View Forum Message <> Reply to Message

On 6/20/12 8:06 AM, alx wrote:
> On 20 juin, 13:01, John Coxon<john.co...@gmail.com> wrote:
>>
>> I see, that would make sense. Is there any way to clear a structure, for
>> instance, between two runs of a program?
>
> If your structure has only numeric fields:
>
> for i=0,N_Tags(your_struct)-1 do your_struct.(i) = 0

This will clear the values, but the definition of avg1. You need to do a
.reset or restart IDL to do that.

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation