Subject: Re: "shrink" structures
Posted by David Fanning on Mon, 08 Feb 2010 13:16:05 GMT
View Forum Message <> Reply to Message

Johannes Korn writes:

>
> Hi,
>
> I usually read ascii tables using read_ascii with a ascii template which
> gives me an anonymous structure with fields of the same length.
>
> for example
> d=READ_ASCII('file', TEMPLATE=templ)
>
> afterwards d is d.date, d.time and d.value
>
> I would like to "shrink" d to contain only a specific date.
>
> Something like
> ind=where(d.date eq '10052000')
> d = d[ind]
>
> This doesn't work of course but I thought at least d.date = d.date[ind]
> should. But this doesn't resize d.date but only fills all locations with
> the values of d.date[ind].
>
> Is there any other way than to copy everything to arrays?

How about a shrunken structure? Consider this function:

```
;************************************************************ ************
Function ShrinkStruct, struct, indices

  tags = N_Tags(struct)
  fieldNames = Tag_Names(struct)

  newStruct = Create_Struct(fieldNames[0], $
                   (struct.(0))[indices])
  FOR j=1,tags-1 DO BEGIN
     newStruct = Create_Struct(newStruct, fieldnames[j], $
                   (struct.(j))[indices])
  ENDFOR

  RETURN, newStruct

END
;************************************************************ ************
```

You can use it like this:

```
 IDL> s = {a:Findgen(101), b:sindgen(101), c:randomu(seed, 101)}
 IDL> indices = Where(s.a gt 60 and s.a lt 75)
 IDL> n = shrinkStruct(s, indices)
 IDL> help, n, /struct
 ** Structure <3918610>, 3 tags, length=336, data length=336, refs=1:
    A               FLOAT     Array[14]
    B               STRING    Array[14]
    C               FLOAT     Array[14]
 IDL> print, n.a
    61.0000  62.0000  63.0000  64.0000  65.0000  66.0000  67.0000
    68.0000  69.0000  70.0000  71.0000  72.0000  73.0000  74.0000
```

Cheers,

David

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

---

## Subject: Re: "shrink" structures
## Posted by wlandsman on Mon, 08 Feb 2010 13:36:21 GMT

View Forum Message <> Reply to Message

David gave a nice solution to this question but I have a couple of additional comments:

On Feb 8, 5:00 am, Johannes Korn <k...@freisingnet.de> wrote:

> 
> I would like to "shrink" d to contain only a specific date.
> 
> Something like<br>
> ind=where(d.date eq '10052000')<br>
> d = d[ind]
>

This would work if you had an array of structures, but not if each tag is an array.  I made the same mistake a couple of weeks ago, when I had a large structure and was trying to save memory by subscripting to the values I needed.   Instead I found my machine crawling to a halt because the line "d = d[ind]" will actually duplicate the structure by the number of values in 'ind', gobbling up all my remaining memory.

> This doesn't work of course but I thought at least d.date = d.date[ind]
> should. But this doesn't resize d.date but only fills all locations with
> the values of d.date[ind].
>

You cannot change the data type or dimensions of a structure tag. That is why David's solution involves creating a whole new structure. (You might want to TEMPORARY() the original structure if your purpose is to save memory.)

But my main problem is with READ_ASCII. The case of reading values into a structure where every tag has the same dimension size really cries out for the use of a structure array (where each element of the structure array corresponds to a "row" of the original ASCII data). I can't think of any drawbacks to this (but I wouldn't be surprised if someone proves me wrong), and it would make things like subscripting the structure much, much easier.

--Wayne

## Subject: Re: "shrink" structures
## Posted by Johannes Korn on Wed, 10 Feb 2010 07:43:11 GMT

View Forum Message <> Reply to Message

David Fanning wrote:
> Johannes Korn writes:
>
>> Hi,
>>
>> I usually read ascii tables using read_ascii with a ascii template which
>> gives me an anonymous structure with fields of the same length.
>>
>> for example
>> d=READ_ASCII('file', TEMPLATE=templ)
>>
>> afterwards d is d.date, d.time and d.value
>>
>> I would like to "shrink" d to contain only a specific date.
>>
>> Something like
>> ind=where(d.date eq '10052000')
>> d = d[ind]
>>
>> This doesn't work of course but I thought at least d.date = d.date[ind]

```
>>  should. But this doesn't resize d.date but only fills all locations with
>>  the values of d.date[ind].
>>
>>  Is there any other way than to copy everything to arrays?
>
>  How about a shrunken structure? Consider this function:
>
>   ;******************************************************** ************
>  Function ShrinkStruct, struct, indices
>
>     tags = N_Tags(struct)
>     fieldNames = Tag_Names(struct)
>
>     newStruct = Create_Struct(fieldNames[0], $
>                            (struct.(0))[indices])
>     FOR j=1,tags-1 DO BEGIN
>        newStruct = Create_Struct(newStruct, fieldnames[j], $
>                               (struct.(j))[indices])
>     ENDFOR
>
>     RETURN, newStruct
>
>  END
>   ;******************************************************** ************
```

That's what I was looking for. Thanks a lot!

Johannes