Subject: Re: Automatic truncation of trailing dimension.....
Posted by Liam E. Gumley on Tue, 27 Nov 2001 21:31:37 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:
>
> .....of an array when the dimension size is 1 is a real pain in the ass. Given:
>
> IDL> x=fltarr(100,1,15)
> IDL> help, x
> X FLOAT = Array[100, 1, 15]
>
> Is there anyway to prevent:
>
> IDL> help, x[*,*,1]
> <Expression> FLOAT = Array[100]
> IDL>
>
> i.e. to give:
> <Expression> FLOAT = Array[100,1]
>
> Argh wot a pain.

I take it you mean

<Expression> FLOAT = Array[100, 1, 1]

Does your code absolutely require it? In many cases, you can use the two interchangeably.

If you must maintain the dimensions:

IDL> dims = size(x, /dimensions)
IDL> index = 1
IDL> help, reform(x[*, *, index], dims[0], dims[1], 1)
<Expression> FLOAT = Array[100, 1, 1]

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

Subject: Re: Automatic truncation of trailing dimension.....
Posted by Liam E. Gumley on Tue, 27 Nov 2001 21:32:01 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:

>
> .....of an array when the dimension size is 1 is a real pain in the ass. Given:
>
> IDL> x=fltarr(100,1,15)
> IDL> help, x
> X FLOAT = Array[100, 1, 15]
>
> Is there anyway to prevent:
>
> IDL> help, x[*,*,1]
> <Expression> FLOAT = Array[100]
> IDL>
>
> i.e. to give:
> <Expression> FLOAT = Array[100,1]
>
> Argh wot a pain.

I take it you mean

<Expression> FLOAT = Array[100, 1, 1]

Does your code absolutely require it? In many cases, you can use the two interchangeably.

If you must maintain the dimensions:

IDL> dims = size(x, /dimensions)
IDL> index = 1
IDL> help, reform(x[*, *, index], dims[0], dims[1], 1)
<Expression> FLOAT = Array[100, 1, 1]

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

Subject: Re: Automatic truncation of trailing dimension.....
Posted by Pavel A. Romashkin on Tue, 27 Nov 2001 21:36:36 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:
>
> IDL> help, x[*,*,1]
> <Expression> FLOAT = Array[100]
> IDL>
>

> i.e. to give:
> <Expression> FLOAT = Array[100,1]

Now this is almost asking for magic. It would be more reasonable to ask for

<Expression> FLOAT = Array[100,1,1] :)

Of course,

s = size(x)
help, reform(x, s[1], 1)

But I think this was brought up so many times that by now it should be
on the request to RSI list. I mean, not truncating the empty dimension.
Those wanting truncation easily could use Reform, or a compile option
could be introduced to toggle this behavior.

Pavel

Subject: Re: Automatic truncation of trailing dimension.....
Posted by Pavel A. Romashkin on Tue, 27 Nov 2001 21:38:33 GMT
View Forum Message <> Reply to Message

Oh, of course should be

s = size(x)
help, reform(x[*,*,1], s[1], 1)

Duh!

"Pavel A. Romashkin" wrote:
>
> Of course,
>
> s = size(x)
> help, reform(x, s[1], 1)

Subject: Re: Automatic truncation of trailing dimension.....
Posted by tam on Tue, 27 Nov 2001 21:45:36 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:
>
> .....of an array when the dimension size is 1 is a real pain in the ass. Given:
>

> IDL> x=fltarr(100,1,15)
> IDL> help, x
> X FLOAT = Array[100, 1, 15]
>
> Is there anyway to prevent:
>
> IDL> help, x[*,*,1]
> <Expression> FLOAT = Array[100]
> IDL>
>
> i.e. to give:
> <Expression> FLOAT = Array[100,1]
>
> Argh wot a pain.
>
> --

Under our previous president I guess I would have said

"I feel your pain"

but now it's

"This is evil."

You might find:

< http://groups.google.com/groups?q=dimension+trailing&hl=
en&group=comp.lang.idl-pvwave&rnum=8&selm=onlngh jin1.fsf%40cow.physics.wisc.edu>

to be helpful.

I suppose you could do something like:

```
sz = size(x)
nvar = reform(x[*,*,1],sz[1],sz[2],1)
help,nvar
```

Everyone seems to be bitten by this at one time or another...

Regards,
Tom McGlynn

Subject: Re: Automatic truncation of trailing dimension.....
Posted by Paul van Delst on Tue, 27 Nov 2001 21:53:06 GMT

"Liam E. Gumley" wrote:
>
> Paul van Delst wrote:
>>
>> .....of an array when the dimension size is 1 is a real pain in the ass. Given:
>>
>> IDL> x=fltarr(100,1,15)
>> IDL> help, x
>> X FLOAT = Array[100, 1, 15]
>>
>> Is there anyway to prevent:
>>
>> IDL> help, x[*,*,1]
>> <Expression> FLOAT = Array[100]
>> IDL>
>>
>> i.e. to give:
>> <Expression> FLOAT = Array[100,1]
>>
>> Argh wot a pain.
>
> I take it you mean
>
> <Expression> FLOAT = Array[100, 1, 1]

No, I mean

<Expression> FLOAT = Array[100,1]

i.e. an rank-2 array where one of the dimensions just happens to be 1. IDL reforms these by default to a vector. This is a pain. Doesn't do it for arrays dimensioned as [1,100], so I don't see why it should do it for the other.

> Does your code absolutely require it?

Not usually - except when I run test cases and I process one instance of something rather than "X" hundred (atmospheric profiles).

> If you must maintain the dimensions:
>
> IDL> dims = size(x, /dimensions)
> IDL> index = 1
> IDL> help, reform(x[*, *, index], dims[0], dims[1], 1)
> <Expression> FLOAT = Array[100, 1, 1]

This is good to know but a pain in the rear to back-implement. To get things working. I just set my minimum allowed dimension to 2 rather than 1. A work-around rather than a fix. I'm not going to bother fixing something that shouldn't be broken - and besides, using IDL is supposed to *enhance* my productivity, not decrease it. :o)

I find this auto-truncate behaviour for arrays with unity dimension in one particular order but not the other quite ridiculous.

paulv

--
Paul van Delst Religious and cultural
CIMSS @ NOAA/NCEP purity is a fundamentalist
Ph: (301)763-8000 x7274 fantasy
Fax:(301)763-8545 V.S.Naipaul

Subject: Re: Automatic truncation of trailing dimension.....
Posted by Paul van Delst on Tue, 27 Nov 2001 22:06:46 GMT
View Forum Message <> Reply to Message

tam wrote:
>
> Paul van Delst wrote:
>>
>> .....of an array when the dimension size is 1 is a real pain in the ass. Given:
>>
>> IDL> x=fltarr(100,1,15)
>> IDL> help, x
>> X FLOAT = Array[100, 1, 15]
>>
>> Is there anyway to prevent:
>>
>> IDL> help, x[*,*,1]
>> <Expression> FLOAT = Array[100]
>> IDL>
>>
>> i.e. to give:
>> <Expression> FLOAT = Array[100,1]
>>
>> Argh wot a pain.
>>
>> --
>
> Under our previous president I guess I would have said
>
> "I feel your pain"
>

> but now it's
>
> "This is evil."

Those darn evil evildoers.

>
> You might find:
>

<Link to Craig Markwardt's tirade against IDL mucking about and changing his data arrays without at least asking first....snipped>

> to be helpful.

Oh, I've read Craig's, uh, comments about this, er, feature in the past.

> Everyone seems to be bitten by this at one time or another...

Well, usually those large chunks missing from my rear-end are my fault. Nuts. The chances of this being FIXED by RSI are probably small since there is more than likely a lot more code that would break if the auto-array-reformat wasn't done. Sigh.

paulv

--
Paul van Delst Religious and cultural
CIMSS @ NOAA/NCEP purity is a fundamentalist
Ph: (301)763-8000 x7274 fantasy
Fax:(301)763-8545 V.S.Naipaul

Subject: Re: Automatic truncation of trailing dimension.....
Posted by R.Bauer on Wed, 28 Nov 2001 01:05:12 GMT
View Forum Message <> Reply to Message

"Pavel A. Romashkin" wrote:
>
> Oh, of course should be
>
> s = size(x)
> help, reform(x[*,*,1], s[1], 1)
>
> Duh!
>
> "Pavel A. Romashkin" wrote:
>>

>> Of course,
>>
>> s = size(x)
>> help, reform(x, s[1], 1)

Dear Pavel,

I have the same problems as you.

Therefore I have written a set routine this is attached.

If RSI won't change the state of removing last dimension 1 I like to
have a
compile option to extend these rule.

Everytime I have defined a three dimensional data set e.g. O3 in LAT and
LON and 1 time
the time is missing and this gives problems if I write HDF or netCDF
datafiles or
if I like to concatinate on the last dimension.

This was one of the reasons I have written dref and set.

It is so terrible if I have once changed my variable by reform and then
all is lost
if I initialize another variable like a=b then a is wrong while b is ok.

regards

Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
http://www.fz-juelich.de/icg/icg1/
 ========== ======
a IDL library at ForschungsZentrum Juelich
 http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml

http://www.fz-juelich.de/zb/text/publikation/juel3786.html
 ========== ======

read something about linux / windows
http://www.suse.de/de/news/hotnews/MS.html

;
;+
; NAME:
;   set
;
; PURPOSE:
;   This functions derefences pointers. If last dimension is 1 this is returned and not the
;   standard IDL Array without last dimension 1.
;   If value is a array of pointer the values are concatinated by concatinate_arrays at the last
dimension
;   If value isn't a pointer this value is returned but with the right dimensions.
;
;
; CATEGORY:
;   PROG_TOOLS
;
; CALLING SEQUENCE:
;   result=set(value,[/free])
;
; INPUTS:
;   value: the pointer to exchange to it's value
;
;
; KEYWORD PARAMETERS:
;   free: if set the pointer is purged from memory
;
; PROCEDURE:
;   This routine should be used if it could be possible that's a value is a value or a pointer
;   of the value. In difference to the idl dereference operator * this routine returns the
;   right dimensions back if last dimension are 1.
;   This function returns concatinated array values if a vector of pointer is submitted.
;   Therefore the dimensions should be conform to do this.
;
; EXAMPLE:
; a=10
; help,set(a) & print,set(a,/free)
; A   INT   =   10
; 10

```
;
;
;  a=ptr_new(10)
;  help,set(a) & print,set(a,/free)
;  A               INT       =       10
;  10
;
;
;
;  a=reform(indgen(10),10,1)
;  X=set(a)
;  y=a
;  help,set(a) & print,set(a,/free) & help, x & help,y
;  A               INT       = Array[10, 1]
;       0       1       2       3       4       5       6       7       8       9
;  X               INT       = Array[10,1]
;  Y               INT       = Array[10]
;
;
;  a=ptr_new(reform(indgen(10),10,1))
;  X=set(a)
;  Y=a
;  help,set(a) & print,set(a,/free) & help, x & help,y
;  A               INT       = Array[10, 1]
;       0       1       2       3       4       5       6       7       8       9
;  X               INT       = Array[10, 1]
;  Y               POINTER   = <PtrHeapVar1388>
;
;
;
;
;
; MODIFICATION HISTORY:
;       Written by:     R.Bauer (ICG-1), 2000-Jul-09
;    2001-11-18 : feature with last dimension eq 1 added
;               : feature concatenate_arrays added: always on last dimension concatination is done
;               : feature added : if value is no pointer this value is returned with the right dimensions
;-


FUNCTION set,value,free=free,_extra=p_key

   IF SIZE(value[0],/TName) EQ 'POINTER' THEN BEGIN

      n_ptr=SIZE(value,/N_ELEMENTS)

      sz=SIZE(*value,/struct)
      IF sz.n_dimensions LE 1 THEN BEGIN
```

```
        IF N_ELEMENTS(result) EQ 0 THEN result=(*value)
        ENDIF ELSE BEGIN
        IF sz.dimensions[sz.n_dimensions-1] EQ 1 THEN BEGIN
          IF N_ELEMENTS(result) EQ 0 THEN
result=REFORM((*value),sz.dimensions[0:sz.n_dimensions-1])
        ENDIF ELSE IF N_ELEMENTS(result) EQ 0 THEN result=(*value)

      ENDELSE
      IF KEYWORD_SET(free) THEN PTR_FREE,value


      RETURN,result
      ENDIF ELSE BEGIN
      sz=SIZE(value,/struct)
      IF sz.n_dimensions GE 2 THEN IF sz.dimensions[sz.n_dimensions-1] EQ 1 THEN BEGIN
        RETURN,REFORM(value,sz.dimensions[0:sz.n_dimensions-1])
      ENDIF ELSE RETURN,value
      RETURN,value
    ENDELSE
END
```

File Attachments

1) set.pro, downloaded 126 times

Subject: Re: Automatic truncation of trailing dimension.....
Posted by Craig Markwardt on Wed, 28 Nov 2001 01:48:46 GMT
View Forum Message <> Reply to Message

tam <tam@lheapop.gsfc.nasa.gov> writes:

> Paul van Delst wrote:
>>
>> .....of an array when the dimension size is 1 is a real pain in the ass. Given:
...
>
> Under our previous president I guess I would have said
>
> "I feel your pain"
>
> but now it's
>
> "This is evil."
>
> You might find:
>
> < http://groups.google.com/groups?q=dimension+trailing&hl=
en&group=comp.lang.idl-pvwave&rnum=8&selm=onlngh jin1.fsf%40cow.physics.wisc.edu>

>
> to be helpful.

What he said, errr, I mean, what I said.

Actually, that particular thread was a little more specialized than
Paul's complaint. Paul, I totally agree with you that dropping the
final unit-dimension is bogus, but the quoted post was premised upon
it happening, and then showing that there were *still* surprises after
that.

For example, the statement "A = DBLARR(N1, N2, N3)" can surprisingly
produce a two- or one-dimensional array!

Or, REFORM() does not work on scalars.

Craig

--
-- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
-- ----------

Subject: Re: Automatic truncation of trailing dimension.....
Posted by Mark Hadfield on Wed, 28 Nov 2001 01:57:44 GMT
View Forum Message <> Reply to Message

From: "Paul van Delst" <paul.vandelst@noaa.gov>
> ...using IDL is supposed
> to *enhance* my productivity, not decrease it. :o)

I tend to find that IDL *diverts* my productivity.

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield
National Institute for Water and Atmospheric Research

--

Subject: Re: Automatic truncation of trailing dimension.....
Posted by David Fanning on Wed, 28 Nov 2001 05:28:28 GMT
View Forum Message <> Reply to Message

Mark Hadfield (m.hadfield@niwa.cri.nz) writes:

> I tend to find that IDL *diverts* my productivity.

Especially when you have to plow through 100+
posts a day. What is this, rec.woodworking!?

Cheers,

David

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