
Subject: Re: Arrays in structures; workarounds?

Posted by [John-David T. Smith](#) on Thu, 04 May 2000 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Craig Markwardt wrote:

>
> "J.D. Smith" <jdsmith@astro.cornell.edu> writes:
>
>> The problem here is reliance on *trailing* shallow dimensions. The
>> IDL manual quotes us:
>>
>> "As with other subscript operations, trailing degenerate dimensions
>> (those with a size of 1) are eliminated."
>>
>> While I can't agree with IDL's mixed notion of a variable's dimensionality
>> between help and direct structure member access, I think IDL has always been
>> clear about this point.
>
> Yes, this aspect of IDL's behavior is documented. I've complained
> about this before because the policy has the habit of biting you at
> very awkward moments. As a programmer, you rarely expect the shapes
> of your arrays to change without your asking!
>
> But my point here was that there is absolutely *no* legitimate way to
> find the true dimensions of a structure tag. Hopefully you will see
> why I want this soon enough, when I have time to finish my little
> project. It's pretty cool.
>
>> The one place you are justified in complaining is the truncation of a single
>> element vector to a scalar: this is a bug (or at least an inconsistency), and
>> affects structure field access only:
>
> Bingo. IDL's behavior of trimming degenerate dimensions from variables
> is *usually* okay because it still leaves you with an array at the
> end. However, when single-element tags in structures are extracted
> they are (a) converted to scalars, and (b) there is no way to know
> that this happened. Try passing this as X or Y to a routine like PLOT
> and you get a crashola. Documented or not, this is ridiculous
> behavior.
>
> Multi-element arrays have a similar problem, but I posted a solution
> for that. I was hoping to close this final gap...

One man's ridiculous behavior is another's feature. I love being able to extract image planes and pass them on to display programs without explicitly chopping off unnecessary dimensions, which I'd bet is why they introduced it in the first place. Do I take it that you're trying to teleport the dimensions over the head of structure extraction and reinstate it post facto? Getting

pretty deperate for those dimensions! When I find it necessary to keep a record of the dimensionality of the object a given data subsection originally belonged to, I arrange my data cubes or hypercubes "on their sides". This obviously won't work for vectors.

We can agree that automatically converting a vector to a scalar is absolutely incorrect, especially since it only seems to happen on structure field extraction.

If you can stand it, use pointers in those structures:

```
IDL> a=[1]
IDL> b={a:ptr_new(a)}
IDL> print,size(*b.a)
      1      1      2      1
```

no dimension chopping there!

JD

--

J.D. Smith	*	WORK: (607) 255-5842
Cornell University Dept. of Astronomy	*	(607) 255-6263
304 Space Sciences Bldg.	*	FAX: (607) 255-5875
Ithaca, NY 14853	*	
