Subject: Re: Automatic truncation of trailing dimension.....
Posted by Paul van Delst on Tue, 27 Nov 2001 21:53:06 GMT
View Forum Message <> Reply to Message

"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