Subject: Re: Problem with array concatenation
Posted by J.D. Smith on Wed, 26 Mar 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:
>
> Hello,
>
> I'm using IDL4.0.1 and I'm having a problem with array concatenation
> that defies any explanation on my part. Here goes....
>
> I'm using the following to code to read in binary data (in a weird
> format) :
>
> LOOP over n_panels....
>
> rad = fltarr(panel_header.n_pts, /nozero)
> tau = fltarr(panel_header.n_pts, /nozero)
>
> readu, lbl_file_lun, rad
> readu, lbl_file_lun, tau
>
> tmp = [[temporary(rad)], [temporary(tau)]]
> if n_panels eq 0 then begin
> lbl_spc = temporary(tmp)
> endif else begin
> lbl_spc = [temporary(lbl_spc), temporary(tmp)]
> endelse
>
> What I want to end up with, in array lbl_spc, is data in the form
> [N, 2] where a total of N points was read in and lbl_spc(*,0) references
> all the "rad" data read in and lbl_spc(*,1) references all the "tau"
> data read in. Typically, panel_header.n_pts is LE 2400 but the total
> number of data points read in after looping a bzillion times can be >
> 5e+06.
>
> Anyway, for the data I'm reading, when I exit my procedure I can get the
> following:
>
> IDL> help, lbl_spc
> LBL_SPC FLOAT = Array(4152834, 2)
>
> which is fine and dandy....right? Well why does this happen:
>
> IDL> help, lbl_spc(*,0)
> help, lbl_spc(*,0)
> ^
> % Syntax error.

>
> And...even more bizarre, I get this:
>
> IDL> help, lbl_spc(0,0)
> <Expression>   STRING   = ''
>
> What gives? My float array suddenly is composed of STRINGS??? The same
> occurs when I use help on any SINGLE element of the array. Any more than
> one element returns a syntax error.
>
> I can retrieve the data if I assign the array to another variable name
> and then everything behaves as it should. Is the method of 2-dimensional
> array concatenation that I'm using a brain-dead way of doing it? BTW, it
> doesn't matter whether I concatenate it as [N,2] or[2,N].
>
> Any help/info/comments/suggestions appreciated.
>
> thanks,
>
> Paul van Delst

I suspect this has to do with one of those pernicious amiguities with IDL's present subscripting syntax (which, thankfully, is changing with IDL 5). What I'd bet is happening is you have a function called lbl_spc somewhere on your !path that is getting compiled. It doesn't even have to be your function... it could just exist is a package of routines you downloaded.

Since IDL has no shadowing protection (i.e. like Mathematica's superb implementation), it doesn't know that lbl_spc(*,0) is not a function call with a malformed argument (*)... so that 'help,lbl_spc(*,0)' gives an error. Also, lbl_spc(0,0) must return an empty string, so that 'help, lbl_spc(0,0)' evaluates help on the string returned by the function.

The only fix is to rename your function or variable (or get IDL 5 and use the [] notation).

JD