Subject: how can i get an [n,1] array without reform?
Posted by mccreigh on Thu, 20 Nov 2008 18:15:36 GMT
View Forum Message <> Reply to Message

This seems so trivial, but I cant figure out anyway of doing this
without using reform and querying the dimensions of the variables,
which i'm afraid will slow me down! :)

in the simplest case, i have two column vectors. everything is fine
when they have more than 1 row.
eg

```
IDL> a=transpose([0,1])
IDL> b=transpose([1,2])
IDL> c=[a,b]
IDL> print,c
      0       1
      1       2
```

My problem comes when a and b shrink to 1 element each.

From the damned if you do department, you can try to concatenate two
row vectors and then transpose:

```
IDL> a=[0]
IDL> b=[2]
IDL> c=[[a],[b]]
IDL> help,c
C               INT       = Array[1, 2]
```

oh, things are looking good... but,

```
IDL> c=transpose(c)
IDL> help,c
C               INT       = Array[2]
```

!!

if there were only some sort of option to transpose that preserved the
number of dimensions. You can specify the dimensions of the result of
transpose, but you have to know them in advance, ie calculate them.

Sorry if I seem obstinate in not wanting to calculate my dimensions.
But you think i could just get a row vector with dims [n,1].

seems like a kind of IDL bar puzzle. or maybe it's easier than that.

Thanks!