Subject: Array index bug....
Posted by steinhh on Sun, 21 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Try this:

```
 data = transpose([[[dist(50)]],[[dist(50)]]],[2,0,1])
 help,data
 ; DATA           FLOAT     = Array[2, 50, 50]

 tvscl,data(0,*,*) ;; Shows the expected image...
```

But, imagine that you're writing a general program that must accept arrays with any number of dimensions, and extract data from the array in (n-1)-dimensional "slices" like the above. Short of having a CASE statement for every extraction of data from the array, the way to do this is e.g.:

```
 tvscl,data(0,*,*,*,*)  ;; Shows the expected image in this case,
                        ;; works for up to 5 dimensions (though
                        ;; I don't know what TVSCL would say about
                        ;; that!)
```

Now, IDL *claims* to handle up to 8 dimensions, so one would think the best thing is to prepare your code for it, e.g.:

```
 tvscl,data(0,*,*,*,*,*,*,*)
```

The image is *not* what you expect! Clearly something breaks down with the eighth dimension.....

Another matter is, how do you create e.g., 4-dimensional data sets with the concatenation style? Lets see:

```
IDL> help,[1,2]
<Expression>    INT       = Array[2]
IDL> help,[[1],[2]]
<Expression>    INT       = Array[1, 2]
IDL> help,[[[1]],[[2]]]
<Expression>    INT       = Array[1, 1, 2]
IDL> help,[[[[1]]],[[[2]]]]

help,[[[[1]]],[[[2]]]]
                 ^
% Only eight levels of variable concatenation are allowed.
```

Hmm. Very strange indeed, since I don't even have 8 opening (or closing) brackets in total, much less anything going to

what *I* would interpret as 8 levels of concatenation....

Regards,

Stein Vidar
