Subject: indexing 2-d vs. 3-d arrays...ARGH!
Posted by Henry on Tue, 19 Apr 2005 21:39:20 GMT
View Forum Message <> Reply to Message

Hey folks,

This one tripped me up today. Wonder how many other places in code
I've been messed up by it. I (vaguely) understand the logic behind it.

Let's say you want to grab some points out of a 2-d image or a 3-d
stack of images:

To set the stage, some pretend images and indexes of the points to
grab:

```
arr2 = dindgen(10,10)
arr3 = dindgen(10,10,5)
index1 = [2,6]
index2 = [3,7]
```

Now,

```
help, arr2[index1,index2]
```

gives

```
<Expression>    DOUBLE    = Array[2]
```

which is what I would expect.

BUT,

```
help,arr3[index1, index2, 2]
```

gives

```
<Expression>    DOUBLE    = Array[2, 2]
```

which is what screwed me up.

IDL demands:

```
help,arr3[index1, index2, replicate(2,2)]
```

to get:

```
<Expression>    DOUBLE    = Array[2]
```

Get it? Got it? Good.

Actually, I take back the comment about vaguely understanding the logic
of the middle case.

-Henry

p.s. Best of luck David. Balance is key...I wish I could figure out
better how to reach it. In 30 years will I appreciate more the hours
spent in my kayak on the lake listening to loons, or the hours beating
my head against the aluminum paneling of my computer? (dang side panel

is too thick to put a satisfying dent in)