Subject: Re: Access array elements with String
Posted by Paul Van Delst[1] on Mon, 14 Jul 2008 19:46:34 GMT
View Forum Message <> Reply to Message

humanumbrella@gmail.com wrote:
> On Jul 14, 11:41 am, Bob Crawford <Snowma...@gmail.com> wrote:
>> On Jul 14, 11:16 am, humanumbre...@gmail.com wrote:
>>
>>
>>
>>> Hello all,
>>> Another issue - perhaps one of you has encountered this before. It's
>>> sort of a neat problem. I'm attempting to build array subscripts on
>>> the fly based on user input. IE the number of static/variable elements
>>> is changing, which allows the user to pick different axes to plot.
>>> Nevermind all that.
>>> Anyway, let's say a user wants a particular axis to be variable. In
>>> this case, the dataset array where I'm attempting to pull values from
>>> would contain a *, to get all these elements. Unfortunately, I do not
>>> know in advance which dimension of the array I will be using, so I am
>>> attempting to build the subscript based on a string.
>>> This was my original thought:
>>> a = dindgen(5,5,5)
>>> b = ['3','3','3']
>>> print, a[b]
>>> but this just returns a[3], a[3], a[3]
>>> So, I figured I'd do it this way:
>>> c = '3'
>>> print, a[c,c,c] -- This works!
>>> Now for the gold,
>>> d = '*'
>>> print, a[c,c,d] -- error - can't convert string-> long
>>> so I get an idea-- maybe I'll just use the ascii value for the
>>> asterisk.
>>> d = String(42b)
>>> print, a[c,d,d] -- error - can't convert string-> long
>>> Any thoughts ?
>>> Thanks in advance
>>> --Justin
>> Why try to force the '*' - might not SIZE be more useful?
>> e.g.
>> s=SIZE(a)
>> print, a[c,c,s[3]] ; for a[c,c,d]
>> print, a[c,s[2],s[3]]; for a[c,d,d]
>
> Hey Bob,
>
> Thanks for the post!

> I think I may need to elaborate a bit more --
> I need the entire row of the multi-dimensional array.
> So, for example, let's say I have an array that is 30 x 20 x 50
> I will need *,0,0 to plot the first 30 values
> but I could just as easily need 0,*,0 or 0,0,* Depending on user
> input, so I can't anticipate that in advance.

What about using execute? I didn't have any problems constructing a string to execute that included the '*' character:

```
pro testit, n
  a=indgen(30,20,50)
  help, a
  info=size(a,/structure)

  index = make_array(info.n_dimensions,value='0')
  index[n] = '*'

  exestring = 'x = reform(a['+strjoin(index,',')+'])'
  result = execute(exestring)
  help, x
end

IDL> testit,0
A               INT       = Array[30, 20, 50]
X               INT       = Array[30]
IDL> testit,1
A               INT       = Array[30, 20, 50]
X               INT       = Array[20]
IDL> testit,2
A               INT       = Array[30, 20, 50]
X               INT       = Array[50]
```

??

cheers,

paulv