Subject: Re: Access array elements with String
Posted by Bob[3] on Mon, 14 Jul 2008 15:41:06 GMT

View Forum Message <> Reply to Message

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]

Subject: Re: Access array elements with String
Posted by humanumbrella on Mon, 14 Jul 2008 15:49:24 GMT
View Forum Message <> Reply to Message

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.

Cheers,
--Justin

Subject: Re: Access array elements with String
Posted by Bob[3] on Mon, 14 Jul 2008 16:30:03 GMT

View Forum Message <> Reply to Message

On Jul 14, 11:49 am, humanumbre...@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.
>
> Cheers,
> --Justin- Hide quoted text -
>
> - Show quoted text -

Oops.
I posted too soon (thank you for the clarification Justin - that is
what I was trying to do)
Here is what I should have posted:

print, a[c,c,0:(s[3]-1)] ; for a[c,c,d]
print, a[c,0:(s[2]-1),0:(s[3]-1)]; for a[c,d,d]

Isn't '*' just short form notation for 0:(s[n]-1), anyway?

Subject: Re: Access array elements with String
Posted by humanumbrella on Mon, 14 Jul 2008 16:59:26 GMT
View Forum Message <> Reply to Message

On Jul 14, 12:30 pm, Bob Crawford <Snowma...@gmail.com> wrote:
> On Jul 14, 11:49 am, humanumbre...@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.
>
>> Cheers,
>> --Justin- Hide quoted text -
>
>> - Show quoted text -
>
> Oops.
> I posted too soon (thank you for the clarification Justin - that is
> what I was trying to do)
> Here is what I should have posted:
>
> print, a[c,c,0:(s[3]-1)] ; for a[c,c,d]
> print, a[c,0:(s[2]-1),0:(s[3]-1)]; for a[c,d,d]
>
> Isn't '*' just short form notation for 0:(s[n]-1), anyway?

Hey Bob,

Yes, I think '*' is short for 0:(s[n]-1) but I read somewhere that you
shouldn't use the range because of performance issues...
Can anyone shed light on that issue?

Subject: Re: Access array elements with String
Posted by humanumbrella on Mon, 14 Jul 2008 17:06:33 GMT

View Forum Message <> Reply to Message

On Jul 14, 12:59 pm, humanumbre...@gmail.com wrote:
> On Jul 14, 12:30 pm, Bob Crawford <Snowma...@gmail.com> wrote:
>
>
>
>> On Jul 14, 11:49 am, humanumbre...@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.
>
>>> Cheers,
>>> --Justin- Hide quoted text -
>
>>> - Show quoted text -
>
>> Oops.
>> I posted too soon (thank you for the clarification Justin - that is
>> what I was trying to do)
>> Here is what I should have posted:
>
>> print, a[c,c,0:(s[3]-1)] ; for a[c,c,d]
>> print, a[c,0:(s[2]-1),0:(s[3]-1)]; for a[c,d,d]
>
>> Isn't '*' just short form notation for 0:(s[n]-1), anyway?
>
> Hey Bob,
>
> Yes, I think '*' is short for 0:(s[n]-1) but I read somewhere that you
> shouldn't use the range because of performance issues...
> Can anyone shed light on that issue?

IE:
From the "Help" pages on "Arrays"
"Processing subscript ranges is inefficient. When possible, use an
array or scalar subscript instead of specifying a subscript range
where the beginning and ending subscripts are separated by the colon
character."

Subject: Re: Access array elements with String
Posted by David Fanning on Mon, 14 Jul 2008 17:09:41 GMT
View Forum Message <> Reply to Message

humanumbrella@gmail.com writes:

> "Processing subscript ranges is inefficient. When possible, use an
> array or scalar subscript instead of specifying a subscript range
> where the beginning and ending subscripts are separated by the colon
> character."

I think article might shed some light:

http://www.dfanning.com/misc_tips/submemory.html

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Access array elements with String
Posted by humanumbrella on Mon, 14 Jul 2008 19:01:18 GMT

View Forum Message <> Reply to Message

On Jul 14, 1:09 pm, David Fanning <n...@dfanning.com> wrote:
> humanumbre...@gmail.com writes:
>> "Processing subscript ranges is inefficient. When possible, use an
>> array or scalar subscript instead of specifying a subscript range
>> where the beginning and ending subscripts are separated by the colon
>> character."
>
> I think article might shed some light:
>
> http://www.dfanning.com/misc_tips/submemory.html
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

It looks like the asterisk character is interpreted as something
different inside the brackets of an array.
This interpretation is not a string '*', so that would explain the
error of trying to use the string in this context.

Is anyone familiar with how to reformat the * in such a way so as the
interpreter would recognize it ?

Thanks in advance,
--Justin

Subject: Re: Access array elements with String
Posted by Bob[3] on Mon, 14 Jul 2008 19:23:40 GMT
View Forum Message <> Reply to Message

On Jul 14, 1:06 pm, humanumbre...@gmail.com wrote:
> On Jul 14, 12:59 pm, humanumbre...@gmail.com wrote:
>
>
>
>
>
>> On Jul 14, 12:30 pm, Bob Crawford <Snowma...@gmail.com> wrote:
>
>>> On Jul 14, 11:49 am, humanumbre...@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.
>
>>>> Cheers,
>>>> --Justin- Hide quoted text -
>
>>>> - Show quoted text -
>
>>> Oops.
>>> I posted too soon (thank you for the clarification Justin - that is
>>> what I was trying to do)
>>> Here is what I should have posted:
>
>>> print, a[c,c,0:(s[3]-1)] ; for a[c,c,d]
>>> print, a[c,0:(s[2]-1),0:(s[3]-1)]; for a[c,d,d]
>
>>> Isn't '*' just short form notation for 0:(s[n]-1), anyway?
>
>> Hey Bob,
>
>> Yes, I think '*' is short for 0:(s[n]-1) but I read somewhere that you
>> shouldn't use the range because of performance issues...
>> Can anyone shed light on that issue?

>
> IE:
> From the "Help" pages on "Arrays"
> "Processing subscript ranges is inefficient. When possible, use an
> array or scalar subscript instead of specifying a subscript range
> where the beginning and ending subscripts are separated by the colon
> character."- Hide quoted text -
>
> - Show quoted text -

It appears from my reading of David's page that quoting the subscripts
by way of using a colon is not any different (memory wise) than using
a '*'.

Given the passage you've quoted above doesn't state that using a '*'
is any more efficient than ':' subsetting, but perhaps doing (as
suggested):

b=INDGEN(s(3))
print, a[c,c,b]

might be more efficient - dunno, haven't tested.

Subject: Re: Access array elements with String
Posted by Jean H. on Mon, 14 Jul 2008 19:44:18 GMT

View Forum Message <> Reply to Message

>
> b=INDGEN(s(3))
> print, a[c,c,b]
>
> might be more efficient - dunno, haven't tested.

You would have to reformat (i.e. repeat the value) c so it has the same
size as b.

Jean

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

Subject: Re: Access array elements with String
Posted by Jean H. on Mon, 14 Jul 2008 19:49:05 GMT

View Forum Message <> Reply to Message

> It looks like the asterisk character is interpreted as something
> different inside the brackets of an array.
> This interpretation is not a string '*', so that would explain the
> error of trying to use the string in this context.
>
> Is anyone familiar with how to reformat the * in such a way so as the
> interpreter would recognize it ?
>
> Thanks in advance,

> --Justin

well... Bob gave you two solutions already
0:s[n] and
indgen(s[n])

Now if you really want to use the *, you might want to have a look at
the command "execute"

```
IDL> a = '*'
IDL> b = indgen(3,3)
IDL> c = '1'
IDL> tmp= execute('e = b[' + c + ',' + a +']')
IDL> print,e
       1
       4
       7
```

but be aware of the limitations of this command (read the help)
Jean

Subject: Re: Access array elements with String
Posted by humanumbrella on Mon, 14 Jul 2008 19:58:45 GMT

View Forum Message <> Reply to Message

On Jul 14, 3:46 pm, Paul van Delst <Paul.vanDe...@noaa.gov> wrote:
> humanumbre...@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

Paulv,

You're the champion of the day! Thanks kindly for your suggestion!

I was unaware of the execute command.

Thanks !!
Cheers,
--Justin

Subject: Re: Access array elements with String
Posted by humanumbrella on Mon, 14 Jul 2008 20:17:02 GMT

View Forum Message <> Reply to Message

On Jul 14, 3:44 pm, Jean H <jghas...@DELTHIS.ucalgary.ANDTHIS.ca> wrote:
>> b=INDGEN(s(3))
>> print, a[c,c,b]
>
>> might be more efficient - dunno, haven't tested.
>
> You would have to reformat (i.e. repeat the value) c so it has the same
> size as b.
>
> Jean

Thanks Bob,

In my particular application, it was difficult to use the solution involving the range, because I had to put ':' in the dataset access function. I do not know how many dimensions are in advance, so I would need to switch on the number of dimensions in order to put the right number of ':' in. I could be missing something,

Thanks Jean for the caution on the execute command, and I will check out the help menu.

Thanks again everyone.
Cheers.
--Justin