Subject: Re: Subscripting arrays.
Posted by Michael Galloy on Thu, 02 May 2013 20:22:52 GMT
View Forum Message <> Reply to Message

On 5/2/13 1:40 PM, rryan@stsci.edu wrote:
> So someone showed me this and I'm trying to wrap my head around it.
> Sorry if it's already posted, I didn't know how to begin to search
> The Group for it...
>
>
> Create an array:
>
> a = intarr(3)
>
> Using IDL 8, subscript with 4 (ie. an index that's too large and
> should be out of bounds)
>
> a[4]=3
>
> Obviously, this crashes.
>
> Conversely subscript with a negative index:
>
> a[-1]=2
>
> does what you expect.
>
> But, now subscript the array, WITH an array:
>
> a[[4]]=2
>
> voila. It works, and simply truncates to the last element of the
> array... Spooky. Now do that with a negative index:
>
> a[[-1]]=4
>
> Like before, it works. But now, it doesn't wrap the index, but
> rather truncates to the first element of the array.
>
> Hmm.. I can't tell if this is the designed behavior, because I have a
> hard time describing it --- it's easier to show it.

Yes, this is the "designed" behavior. You can turn off the odd array indexing (and just give an error) with compile_opt:

```
IDL> x = findgen(10)
IDL> print, x[[-1, 0, 9, 10]]
      0.00000      0.00000      9.00000      9.00000
```

IDL> compile_opt strictarrsubs
IDL> print, x[[-1, 0, 9, 10]]
% Array used to subscript array contains out of range subscript: X.
% Execution halted at: $MAIN$

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation

---

Subject: Re: Subscripting arrays.
Posted by Russell Ryan on Fri, 03 May 2013 20:09:57 GMT
View Forum Message <> Reply to Message

A ha. Sounds reasonable, but I'm not sure why it's "designed" this way, but c'est la vie. I passed your response on to the friend, who groaned. We nearly said (in unison), I never pay much attention to those compiler options.

Thanks!
Russell

On Thursday, May 2, 2013 4:22:52 PM UTC-4, Mike Galloy wrote:
> On 5/2/13 1:40 PM, u wrote:
>
>> So someone showed me this and I'm trying to wrap my head around it.
>
>> Sorry if it's already posted, I didn't know how to begin to search
>
>> The Group for it...
>
>>
>
>>
>
>> Create an array:
>
>>
>
>> a = intarr(3)
>
>>
>
>> Using IDL 8, subscript with 4 (ie. an index that's too large and
>

>> should be out of bounds)
>
>>
>
>> a[4]=3
>
>>
>
>> Obviously, this crashes.
>
>>
>
>> Conversely subscript with a negative index:
>
>>
>
>> a[-1]=2
>
>>
>
>> does what you expect.
>
>>
>
>> But, now subscript the array, WITH an array:
>
>>
>
>> a[[4]]=2
>
>>
>
>> voila. It works, and simply truncates to the last element of the
>
>> array... Spooky. Now do that with a negative index:
>
>>
>
>> a[[-1]]=4
>
>>
>
>> Like before, it works. But now, it doesn't wrap the index, but
>
>> rather truncates to the first element of the array.
>
>>
>

>> Hmm.. I can't tell if this is the designed behavior, because I have a
>
>> hard time describing it --- it's easier to show it.
>
>
>
> Yes, this is the "designed" behavior. You can turn off the odd array
>
> indexing (and just give an error) with compile_opt:
>
>
>
> IDL> x = findgen(10)
>
> IDL> print, x[[-1, 0, 9, 10]]
>
>  0.00000 0.00000 9.00000 9.00000
>
> IDL> compile_opt strictarrsubs
>
> IDL> print, x[[-1, 0, 9, 10]]
>
> % Array used to subscript array contains out of range subscript: X.
>
> % Execution halted at: $MAIN$
>
>
>
> Mike
>
> --
>
> Michael Galloy
>
> www.michaelgalloy.com
>
> Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
>
> Research Mathematician
>
> Tech-X Corporation

---