Subject: Re: Array indexing problem
Posted by David Fanning on Wed, 28 Jan 2004 21:34:03 GMT
View Forum Message <> Reply to Message

Roy writes:

> Is there any existing code that will catch the below IDL simple
> example? Input a and b must be an array.
>
>
> Input is an array:
> IDL> b=[4]
> IDL> a=[0,1,2]
> IDL> print,a[b]
> 2
> This should give me an error.

Alas, you have run into a little known, but enormously
appreciated, little feature of IDL. (I'm sure it is
explained in the manuals somewhere. :-)

Maybe something like this for an error handler:

```
  IF Total(b GT ((N_Elements(a) -1)) GT 0 THEN $
    Message, 'B indices exceed bounds of A'
```

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Array indexing problem
Posted by hunter on Wed, 28 Jan 2004 21:43:42 GMT
View Forum Message <> Reply to Message

You can also turn off this feature if it offends.

compile_opt strictarrsubs

And you will get an error for the example you mentioned above.

Eli

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.1a81d8b95abba27f9897bd@news.frii.com...
> Roy writes:
>
>> Is there any existing code that will catch the below IDL simple
>> example? Input a and b must be an array.
>>
>>
>> Input is an array:
>> IDL> b=[4]
>> IDL> a=[0,1,2]
>> IDL> print,a[b]
>> 2
>> This should give me an error.
>
> Alas, you have run into a little known, but enormously
> appreciated, little feature of IDL. (I'm sure it is
> explained in the manuals somewhere. :-)
>
> Maybe something like this for an error handler:
>
> IF Total(b GT ((N_Elements(a) -1)) GT 0 THEN $
> Message, 'B indices exceed bounds of A'
>
> Cheers,
>
> David
> --
> David W. Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Phone: 970-221-0438, E-mail: david@dfanning.com
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Array indexing problem
Posted by Liam Gumley on Wed, 28 Jan 2004 22:11:45 GMT
View Forum Message <> Reply to Message

"Roy" <roberson_1@yahoo.com> wrote in message
news:c0a9aee5.0401281200.400888f1@posting.google.com...
> Is there any existing code that will catch the below IDL simple
> example? Input a and b must be an array.
>
> Input is an array:

> IDL> b=[4]
> IDL> a=[0,1,2]
> IDL> print,a[b]
>   2
> This should give me an error.

This is how array subscripting works in IDL, when the subscript is an array. To quote the IDL Manual "Building IDL Applications", Chapter 5, "Using Arrays as Subscripts":

"If an element of the subscript array is less than or equal to zero, the first element of the subscripted variable is selected. If an element of the subscript is greater than or equal to the last subscript in the subscripted variable (N, above), the last element is selected."

If the subscript variable has just one elements, then you can do this instead

IDL> print, a[b[0]]
% Attempt to subscript A with <LONG   (4)> is out of range.
% Execution halted at: $MAIN$

since b[0] is a scalar. You can do this even if b is a scalar.

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

Subject: Re: Array indexing problem
Posted by Michael Wallace on Wed, 28 Jan 2004 22:23:10 GMT
View Forum Message <> Reply to Message

>> Is there any existing code that will catch the below IDL simple
>> example? Input a and b must be an array.
>>
>>
>> Input is an array:
>> IDL> b=[4]
>> IDL> a=[0,1,2]
>> IDL> print,a[b]
>>   2
>> This should give me an error.
>
>
> Alas, you have run into a little known, but enormously
> appreciated, little feature of IDL. (I'm sure it is

> explained in the manuals somewhere. :-)

Could you elaborate on the "enormously appreciated" part of your statement? Maybe I'm not thinking in the "IDL Way," but I would not have expected the result IDL gives. It may very well be explained in the manuals, but why is such a "feature" "enormously appreciated?"

Mike

Subject: Re: Array indexing problem
Posted by Paul Van Delst[1] on Wed, 28 Jan 2004 22:28:22 GMT
View Forum Message <> Reply to Message

Michael Wallace wrote:
>
>>> Is there any existing code that will catch the below IDL simple
>>> example? Input a and b must be an array.
>>>
>>>
>>> Input is an array:
>>> IDL> b=[4]
>>> IDL> a=[0,1,2]
>>> IDL> print,a[b]
>>> 2
>>> This should give me an error.
>>
>>
>> Alas, you have run into a little known, but enormously
>> appreciated, little feature of IDL. (I'm sure it is
>> explained in the manuals somewhere. :-)
>
> Could you elaborate on the "enormously appreciated" part of your
> statement? Maybe I'm not thinking in the "IDL Way," but I would not
> have expected the result IDL gives. It may very well be explained in
> the manuals, but why is such a "feature" "enormously appreciated?"

After reading Liam's explanation, I would _still_ like elaboration on why this is "enormously appreciated"! :o) Seems a bizarre feature to me..... but at least it's documented.

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC

Subject: Re: Array indexing problem
Posted by David Fanning on Wed, 28 Jan 2004 22:40:12 GMT
View Forum Message <> Reply to Message

Michael Wallace writes:

> Could you elaborate on the "enormously appreciated" part of your
> statement? Maybe I'm not thinking in the "IDL Way," but I would not
> have expected the result IDL gives. It may very well be explained in
> the manuals, but why is such a "feature" "enormously appreciated?"

I can't remember. :-)

I do remember the first time I pointed this "bug"
out on the IDL newsgroup years and years ago and
how mortified I was when Ray Sterner publicly
pointed out what a GREAT and USEFUL thing it was.
(In those days Ray was the word on the IDL newsgroup.)

In fact, it may have been this fiasco that caused
Dave Stern to send around that memo forbidding RSI
employees from posting on the newsgroup.

Odd that in 17 years of programming I've never run
across the need for this. But there you go. An historical
perspective.

Maybe we can hear from someone who uses it.

Cheers,

David

P.S. Let's just say I've never bothered to sing much
either after my 7th grade music teacher in a fit of pique
told me I had a "two note range". Funny what power those
formative experiences have on you. :-)

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Array indexing problem
Posted by Mark Hadfield on Thu, 29 Jan 2004 00:55:41 GMT

David Fanning wrote:
> Maybe we can hear from someone who uses it.

Not me, but I would like to point out that this "desirable" behaviour
can be eliminated in version 6.0 by adding the following declaration to
your code

  compile_opt STRICTARRSUBS

I have this, along with DEFINT32, STRICTARR & LOGICAL_PREDICATE in my
start-up file and all recently-edited routines.

This done, Roy's example gives me

% Array used to subscript array contains out of range subscript: A.

--
Mark Hadfield "Ka puwaha te tai nei, Hoea tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: Array indexing problem
Posted by roberson_1 on Thu, 29 Jan 2004 17:03:42 GMT

Thank you for responding so quickly. Compile_opt STRICTARRSUBS works
for IDL 6.0 but I have to use 5.1. Is there a way to add in a compile
function, or some else that will look for this error in the original
code? I don't want to add an error handler, which David suggested.
The original code can not be altered.

Roy

"Liam Gumley" <pip_book@mailinator.com> wrote in message
news:<bv9c35lqf1@news.doit.wisc.edu>...
> "Roy" <roberson_1@yahoo.com> wrote in message
> news:c0a9aee5.0401281200.400888f1@posting.google.com...
>> Is there any existing code that will catch the below IDL simple
>> example? Input a and b must be an array.
>>
>> Input is an array:
>> IDL> b=[4]
>> IDL> a=[0,1,2]
>> IDL> print,a[b]
>> 2

>> This should give me an error.
>
> This is how array subscripting works in IDL, when the subscript is an array.
> To quote the IDL Manual "Building IDL Applications", Chapter 5, "Using
> Arrays as Subscripts":
>
> "If an element of the subscript array is less than or equal to zero, the
> first element of the subscripted variable is selected. If an element of the
> subscript is greater than or equal to the last subscript in the subscripted
> variable (N, above), the last element is selected."
>
> If the subscript variable has just one elements, then you can do this
> instead
>
> IDL> print, a[b[0]]
> % Attempt to subscript A with <LONG (4)> is out of range.
> % Execution halted at: $MAIN$
>
> since b[0] is a scalar. You can do this even if b is a scalar.
>
> Cheers,
> Liam.
> Practical IDL Programming
> http://www.gumley.com/

Subject: Re: Array indexing problem
Posted by Liam Gumley on Thu, 29 Jan 2004 17:43:09 GMT
View Forum Message <> Reply to Message

I think your options are

(1) Modify the code

or

(2) Use IDL 6.0

Perhaps you could show us the original code?

"Roy" <roberson_1@yahoo.com> wrote in message
news:c0a9aee5.0401290903.154044d3@posting.google.com...
> Thank you for responding so quickly. Compile_opt STRICTARRSUBS works
> for IDL 6.0 but I have to use 5.1. Is there a way to add in a compile
> function, or some else that will look for this error in the original
> code? I don't want to add an error handler, which David suggested.
> The original code can not be altered.
>

> Roy
>
> "Liam Gumley" <pip_book@mailinator.com> wrote in message
news:<bv9c35lqf1@news.doit.wisc.edu>...
>> "Roy" <roberson_1@yahoo.com> wrote in message
>> news:c0a9aee5.0401281200.400888f1@posting.google.com...
>>> Is there any existing code that will catch the below IDL simple
>>> example? Input a and b must be an array.
>>>
>>> Input is an array:
>>> IDL> b=[4]
>>> IDL> a=[0,1,2]
>>> IDL> print,a[b]
>>> 2
>>> This should give me an error.
>>
>> This is how array subscripting works in IDL, when the subscript is an
array.
>> To quote the IDL Manual "Building IDL Applications", Chapter 5, "Using
>> Arrays as Subscripts":
>>
>> "If an element of the subscript array is less than or equal to zero, the
>> first element of the subscripted variable is selected. If an element of
the
>> subscript is greater than or equal to the last subscript in the
subscripted
>> variable (N, above), the last element is selected."
>>
>> If the subscript variable has just one elements, then you can do this
>> instead
>>
>> IDL> print, a[b[0]]
>> % Attempt to subscript A with <LONG (4)> is out of range.
>> % Execution halted at: $MAIN$
>>
>> since b[0] is a scalar. You can do this even if b is a scalar.
>>
>> Cheers,
>> Liam.
>> Practical IDL Programming
>> http://www.gumley.com/

Subject: Re: Array indexing problem
Posted by David Fanning on Thu, 29 Jan 2004 17:58:01 GMT
View Forum Message <> Reply to Message

Roy writes:

> Thank you for responding so quickly. Compile_opt STRICTARRSUBS works
> for IDL 6.0 but I have to use 5.1. Is there a way to add in a compile
> function, or some else that will look for this error in the original
> code? I don't want to add an error handler, which David suggested.
> The original code can not be altered.

I think this request falls into the general category of wanting
to have your cake and eat it too. :-)

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Array indexing problem
Posted by kashyap on Thu, 29 Jan 2004 23:21:00 GMT
View Forum Message <> Reply to Message

In article <MPG.1a81e83474fc47e19897be@news.frii.com>,
David Fanning <david@dfanning.com> wrote:
> Michael Wallace writes:
>
> Maybe we can hear from someone who uses it.
>
> Cheers,
>
> David

I use this feature quite a lot, for instance, to
define an array of a different size using a set
of indices where said indices may overflow (happens
all the time), and all I may want at these overflow
points is some default value.

Example: suppose there is an array of element
abundances (say ABUND, with size 30). And a
set of indices indicating the atomic number
(say Z, of size 10386). I want to make a new
array (AZ, of size 10386) which contains the
abundances for each of the Z[i]. Now further

suppose that in some places Z may be 0 or -1,
and for which I simply want to set the abundance
to 1. There are no doubt other ways of doing
this, but AZ=ABUND[Z-1] works fast, transparently,
AND stays readable.

Vinay
--

_____
_____
kashyap@head-cfa.harvard.edu 617 495 7173 [CfA/P-143] 617 496 7173 [F]