
Subject: last array index subscript

Posted by [Chad Bender](#) on Thu, 23 Jan 2003 17:50:30 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi --

Is there a way in IDL to directly reference the last element of a vector without first determining how many elements the vector contains?

For example, something like:

```
n=N_Elements(my_array)
value=my_array[n-1]
```

except without having to make the call to N_Elements. Granted, avoiding the N_Elements call probably doesn't save a lot of time. But I figured that with all of the complicated syntax that IDL accepts, there might be a direct way to do this. The Bldg Apps guide says you can extract a subarray from some element e to the end of the array like this:

```
subarray=my_array[e:*
```

So it seems to me that IDL knows what that last index is. I tried something like:

```
value=my_array[*-1], but (not unexpectedly) this caused a syntax error.
```

I figure that what I want is probably impossible, but if anyone knows how to do it your advice is appreciated.

Thanks
Chad Bender

Subject: Re: last array index subscript

Posted by [kashyap](#) on Fri, 31 Jan 2003 23:35:49 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Doug,

All I know is that it is a feature, not a bug. Subscripting a variable with an array of indices returns an array of the same size. Where the index numbers undershoot (-ve values) the output gets filled with the first element and where the index numbers overshoot, those get filled out with the last element.

Look under the "Array Subscripts" section of the online help.

Vinay

In article <20030131103101140-0600@news.tc.umn.edu>,

Doug Rowland <rowland@fields.space.umn.edu> wrote:

> Vinay,

>

> How does this work? I tried to wrap my brain around this one. Is it an

> IDL "bug" or is there some simple rationale for IDL behaving this way?

> For example, if I try to directly subscript my_array with the scalar

> value 2147483647L (or any other number which is larger than the number

> of elements of my_array, less one) I get

>

> IDL> print,my_array[2147483647L]

> % Attempt to subscript A with <LONG (2147483647)> is out of range.

> % Execution halted at: \$MAIN\$

>

> Why should wrapping this index value in an array suddenly make it work?

>

> Thanks.

>

> Doug

>

> In <3e304dac.0@cfanews.harvard.edu> Vinay L. Kashyap wrote:

>> Try

>>

>> value=(my_array[[2147483647L]])[0]

>>

>> where the "[.]" returns an array and the "(.)[0]" ensures that the

>> output is a scalar, and $2147483647L = 2L^{(31L)} - 1L$ is the largest

>> possible $l*4$ number you can have and surely no will have an array

>> bigger than that.

>>

>> vinay

--

kashyap@head-cfa.harvard.edu

617 495 7173 [CfA/P-143] 617 496 7173 [F]

Subject: Re: last array index subscript

Posted by [Mark Hadfield](#) on Sun, 02 Feb 2003 05:20:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

"Vinay L. Kashyap" <kashyap@head-cfa.harvard.edu> wrote in message
news:3e3b0855.0@cfanews.harvard.edu...

> All I know is that it is a feature, not a bug.

Such a worthwhile feature that in IDL 5.6 a `COMPILE_OPT` keyword was introduced to disable it!

`STRICTARRSUBS` - When IDL subscripts one array using another array as the source of array indices, the default behavior is to clip any out-of-range indices into range and then quietly use the resulting data without error. This behavior is described in `Array Subscripting`. Specifying `STRICTARRSUBS` will instead cause IDL to treat such out-of-range array subscripts within the body of the routine containing the `COMPILE_OPT` statement as an error. The position of the `STRICTARRSUBS` option within the module is not important: All subscripting operations within the entire body of the specified routine will be treated this way.

--

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