
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 I^*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]
