Subject: Re: Unexpected array subscript truncation
Posted by Michael Galloy on Mon, 30 Nov 2009 22:01:07 GMT
View Forum Message <> Reply to Message

M. Katz wrote:

```
> Here's an interesting little array trick I discovered by accident. It
> doesn't give an error message where I would expect one! Try this:
>
> * As a warm-up enter this and see that IDL does complain.
>
> a = intarr(3,3)
> a[0,5] = 1
>
> * Now try to set it as follows, with an array of y-index values.
>
> a[0,[0,5]] = 1
> print, a
>       1      0      0
>       0      0      0
>       1      0      0
>
> No complaint! In fact, IDL quietly truncated the 5 down to its highest
> allowed value, which is 2.
>
> It also works like this:
>
> a[*,[0,9]] = 3
>
> is equivalent to
>
> a[*,[0,2]] = 3
>
> The same truncation happens with x indices as well.
>
> I suppose there could be some cases where this would not be the
> desired behavior. Any thoughts? Am I missing something?
```

Note that IDL will not only bring too large subscript back the last element, but also will bring negative subscripts back to 0. This is the case any time you index an array by an array of indices.

You can make IDL complain instead of quietly truncating by setting the strictarrsubs compile option:

```
  compile_opt strictarrsubs
```

Mike
--

www.michaelgalloy.com
Research Mathematician
Tech-X Corporation