Subject: Re: Negative indexing and the WHERE function in IDL 8.0
Posted by wlandsman on Thu, 19 Aug 2010 13:29:10 GMT

View Forum Message <> Reply to Message

On Aug 19, 8:09 am, svhhaugan <s.v.h.hau...@gmail.com> wrote:
> On Aug 18, 6:31 pm, Chris Torrence <gorth...@gmail.com> wrote:
>
> How about using "compile_opt strictarrsubs" (which is IDL 7-
> compatible)
> to imply what it always implied in IDL 7 and before: all negative
> array subscripts are illegal, whether scalar or arrays.

That won't work for IDL 8.0 users who want to use the negative scalar indexing, but still need strictarrsubs to give an error when subscripting an array with another array with out of bound indices.

```
IDL> print,!version
{ x86_64 darwin unix Mac OS X 8.0 Jun 17 2010      64      64}

IDL> a = intarr(4)
IDL> a[ [-1,1,2,5] ] = [6,7,8,9]
IDL> print,a
       6       7       8       9
```

--Wayne

P.S.   I go back to the very early days (1986) of IDL when WHERE() didn't have the COUNT parameter, and one had to specifically test whether the returned index was equal to -1.

```
index = where(array)
if index[0] NE -1 then ....
```

but I had never seen before the use of CATCH to make sure WHERE() returns a valid value.   It seems very convoluted to me, but I suppose it makes sense if one expects WHERE() to normally return a valid index, and a single CATCH clause can be used for multiple WHERE() statements to test for errors.   But now those errors won't be caught...