
Subject: Re: Warning: IDL 8.0 alters the behaviour of existing valid programs without any notice!

Posted by [Maarten\[1\]](#) on Wed, 18 Aug 2010 12:14:15 GMT

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

On Aug 18, 1:59 pm, svhhaugan <s.v.h.hau...@gmail.com> wrote:

```
> pro test,data
>   catch,error
>   if error ne 0 then begin
>     catch,/cancel
>     print,"Guess that didn't work"
>     return
>   end
>   data[where(data eq min(data)-1)] = 50
> end
>
> Perfectly valid code, makes sense in IDL 7.1 although the catch,error
> part would most likely be done "with human intervention".
```

I just looked up the help for where in IDL 7. This is what is prominently displayed on that page:

Note: When WHERE Returns -1

If all the elements of Array_Expression are zero, WHERE returns a scalar integer with a value of -1. Attempting to use this result as an index into another array results in a "subscripts out of bounds" error. In situations where this is possible, code similar to the following can be used to avoid errors:

```
; Use Count to get the number of nonzero elements:
index = WHERE(array, count)
```

```
; Only subscript the array if it's safe:
IF count NE 0 THEN result = array[index]
```

And yes, you can use the try/catch exception method of handling this, but I would pose that testing for an item not found message and then using an IF statement to handle that common situation makes more sense than relying on an error. So in that sense the code was never valid or sensible in the first place.

Does that mean no one will be caught off-guard by this? No. Is that an issue? Some people just get what they deserve. The added benefit of being able to count from the end of an array is certainly worth some trouble.

Maarten
