
Subject: Re: N_ELEMENTS and WHERE: Scalar or Array ?

Posted by [wmc](#) on Mon, 01 Feb 1999 08:00:00 GMT

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

rmlongfield@my-dejanews.com wrote:

> Hi All, I keep making the same mistake with N_ELEMENTS so I decided
> to write and ask if anyone has found a solution. I use WHERE to find some
> zeroes in a data set which I want to exclude in further processing. Problem
> is that sometimes they are all zeroes. Using a simple :

> non_zero_xvalues = WHERE (subarray1 GT 0)
> IF(non_zero_array EQ -1) ...

> gets me into trouble because, when it is an array, I get an error. If I use
> N_ELEMENTS(non_zero_array) there is always at least one element, whether it
> is -1 or something else. I don't like ignoring the 'something else' value
> just because it is the only one. Is the answer another IF statement or some
> sort of error control?

The temporary solution to this is to use where(wurble,count)...

But since this has come up, & its one of my pet peeves: why cannot where
return a null array to indicate no-elements-match. And then array[null] would
match to nothing. This would allow one to say

```
array[where(wurble)]= 'stoat'
```

instead of the ugly

```
i=where(wurble,count)  
if (count gt 0) then array[i]= 'stoat'
```

I use Perl a lot, and the contrast is very striking there: perl handles null
values quite happily and it simplifies a lot of things.

While I'm here: would RSI please put a decent regexp package into IDL?

-W.

--

William M Connolley | wmc@bas.ac.uk | <http://www.nbs.ac.uk/public/icd/wmc/>
Climate Modeller, British Antarctic Survey | Disclaimer: I speak for myself
