Subject: Interesting WHERE function gotcha
Posted by David Fanning on Fri, 07 Feb 2003 02:23:40 GMT
View Forum Message <> Reply to Message

Folks,

I just spent a pleasant hour or so chasing down an interesting WHERE function gotcha. I thought you might be interested.

I have an alphabetized string array:

```
array = ['apple', 'avacado', 'banana', 'carrot']
```

I wish to make a list of those vegetables (I think of them as vegetables) that begin with the letter "a". I want this to be fast (there are several hundreds of entries in my real array), so I plan to search for byte values.

```
index = WHERE( (Byte(array))[0,*] EQ Byte('a'), count)
Print, count
   1
```

Uh, huh. (Bit of head scratching here.)

I probably did the extraction incorrectly. Try this:

```
veggie_letter = (Byte(array))[0,*]
Print, Reform(veggie_letter)
  97  97  98  99
letter = Byte('a')
Print, letter
  97
```

Uh, huh. Let's see, "One, two 97s in there." Well, that's interesting. :-(

How about this:

```
Help, veggie_letter, letter
  VEGGIE_LETTER   BYTE      = Array[4]
  LETTER          BYTE      = Array[1]
```

"LETTER, a byte *array*!? You don't suppose..." Try this:

```
index = WHERE( (Byte(array))[0,*] EQ (Byte('a'))[0], count)
Print, count
   2
```

Hummm. V-e-r-y interesting...

Now I know how to fix the problem, but I don't know exactly
what the problem is. (Although this is not so different
from most computer problems, when you come to think of it.)
Is the problem that the BYTE function always makes a byte
array when extracting string arguments? Or is it that
the WHERE function acts in a, uh, non-intuitive way when
there are two vectors in a boolean expression?

And how *does* this WHERE expression work, anyway? Why
don't I get errors? How can I exploit a boolean expression
involving two vectors?

As usual, more questions than answers when you look deeper.
Any ideas? :-)

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Interesting WHERE function gotcha
Posted by JD Smith on Mon, 10 Feb 2003 18:28:56 GMT
View Forum Message <> Reply to Message

On Mon, 10 Feb 2003 10:21:46 -0700, Stein Vidar Hagfors Haugan wrote:

> JD Smith <jdsmith@as.arizona.edu> writes:
>
>> On Thu, 06 Feb 2003 19:35:31 -0700, Craig Markwardt wrote:
> [...]
>> Yes, the tension between single element arrays and scalars is felt
>> deeply, even within the core IDL development group. It's a regrettable
>> legacy which must, unfortunately, be preserved for compatibitlity.
>> There are, however, minor changes and improvements: e.g., with v5.6,
>> KEYWORD_SET now considers a single element array TRUE only if its
>> single element is non-zero; i.e. it's treated just like a scalar for
>> this (and only this) function.
>

> Uh... improvement? I beg to differ. Not only does it introduce a single
> mysterious case where the legacy compatibility fails (when someone
> relies on the previous truth that "when it's defined as an array, it's
> set!") (I hope I never have to debug a code experiencing this!), but it
> introduces even one more level of exception to how singular arrays are
> treated.... Okay, so trailing singular dimensions are disappearing or
> are ignored, except when it's the last dimension to survive the
> slaughter, *except* that this one nifty function that we wrote
> specifically to say that anything not a scalar zero means "set" is right
> now doing a vote on whether to consider your singular dimension worthy
> of noticing... Ugh..!

Notice I said "changes and improvements". You pick which is applicable ;). The truth is, I was once told by a top RSI developer, "If I were designing IDL over, there would be no scalars, just arrays of various dimensionality." Given that we can't go back there, I'm not sure which is better, lots of little workarounds, or just living with the pain. I for one tend to throw a lot of [0] indexing statements in for good measure.

JD

Subject: Re: Interesting WHERE function gotcha
Posted by Craig Markwardt on Tue, 11 Feb 2003 04:24:32 GMT
View Forum Message <> Reply to Message

JD Smith <jdsmith@as.arizona.edu> writes:
>
> Notice I said "changes and improvements". You pick which is
> applicable ;). The truth is, I was once told by a top RSI developer,
> "If I were designing IDL over, there would be no scalars, just arrays
> of various dimensionality." Given that we can't go back there, I'm
> not sure which is better, lots of little workarounds, or just living
> with the pain. I for one tend to throw a lot of [0] indexing
> statements in for good measure.

I think of scalars and one element arrays quite differently. I'm glad IDL keeps them separate. Actually, in a parallel universe where scalars didn't exist, I'm sure we'd all be complaining about something else.

Alter-discussion:

JC Smith: "Research Cisterns Incorporated has just added something new called scalars!"

Stein Hagdorf: "Oh, no, that just adds another exception to all my

array processing! SOHO will crash!"

Clyde Markwardt: "Gosh darnit! REFORM won't work on those new scalars. Can't they fix the old stuff before adding new stuff?"

David Franning: "Well, at least typing less of those []'s is going to make my tennis elbow better!"

Yours,
Clyde

[P.S. REFORM still doesn't work on scalars.]

--

---------- ----------
Clyde B. Markwardt, Ph.D. EMAIL: clydemnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------