Subject: Where
Posted by silviadncl on Tue, 28 Oct 2008 09:48:31 GMT
View Forum Message <> Reply to Message

Hallo!
Why IDL returns -1 when i do:
where(array eq 0.785)?
Array elements are not zero!
Thank you!

Subject: Re: WHERE
Posted by Kenneth P. Bowman on Fri, 05 Nov 2010 13:52:13 GMT
View Forum Message <> Reply to Message

In article
<8c77484b-ac8e-4579-ab69-347ff498411f@k14g2000pre.googlegroups.com>,
Paul <paulstaten@gmail.com> wrote:

> Now that -1 is a valid array index, does it make sense for WHERE to
> return it as a null result?

If you are not in the habit of always checking the 'count'
returned by WHERE, now is the time to start. :-)

Oh, and update your legacy code also.

Ken Bowman

Subject: Re: WHERE
Posted by David Fanning on Fri, 05 Nov 2010 14:01:39 GMT
View Forum Message <> Reply to Message

Paul writes:

> Now that -1 is a valid array index, does it make sense for WHERE to
> return it as a null result?

This is a dilemma, because it certainly should return a NULL
result. But if you force it to, you suddenly lose a lot
of backward compatibility and the ability to write
platform independent programs.

This might be one of those IDL tattoos. You just
have to learn to live with the regret engendered
by some of the foolish things you did in your youth.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: WHERE
Posted by penteado on Fri, 05 Nov 2010 15:20:21 GMT
View Forum Message <> Reply to Message

On Nov 5, 11:44 am, Paul <paulsta...@gmail.com> wrote:
> Now that -1 is a valid array index, does it make sense for WHERE to
> return it as a null result?

This was discussed at length last year, before the choices in 8.0 were made. It absolutely has to keep returning -1 if called the same way, or it would break a lot of old code - many tests are made comparing the result with -1, instead of using the count argument. Living with the consequences of the past, as David said.

New code can make use of the (new, obviously) null keyword, that makes it return !null instead of -1. And the use of !null as index was made to go nicely with that, so that doing assigning with a !null index does nothing, with no errors, and retrieving with a !null index returns !null, also with no errors.

Subject: Re: WHERE
Posted by on Fri, 05 Nov 2010 15:31:13 GMT
View Forum Message <> Reply to Message

On Nov 5, 4:20 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
> It absolutely has to keep returning -1 if called the same way,

Fully agree with David/Paulo!

And what about somehow allowing a Null result starting with IDL 8.0?
Such as a new keyword for that, compile_opt, or whatever the best way
to do this is...

As you write, the null result in WHERE has many advantages as it can
be changed to assign something later without getting an error. It
would be a pity not being able to exploit this.

Subject: Re: WHERE
Posted by on Fri, 05 Nov 2010 15:36:55 GMT
View Forum Message <> Reply to Message

On Nov 5, 4:31 pm, Axel M <axe...@gmail.com> wrote:
> On Nov 5, 4:20 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
>
>> It absolutely has to keep returning -1 if called the same way,
>
> Fully agree with David/Paulo!
>
> And what about somehow allowing a Null result starting with IDL 8.0?
> Such as a new keyword for that, compile_opt, or whatever the best way
> to do this is...
>
> As you write, the null result in WHERE has many advantages as it can
> be changed to assign something later without getting an error. It
> would be a pity not being able to exploit this.

I just saw that there IS a /NULL keyword. Sorry for the post then...

Subject: Re: WHERE
Posted by Paul[3] on Fri, 05 Nov 2010 18:51:12 GMT
View Forum Message <> Reply to Message

Ah, I hadn't checked for the NULL keyword...I'd wished for it in
passing, so thanks for the discussion! I'll pass that on to my friends
here in the department.

Subject: Re: WHERE
Posted by Paul Van Delst[1] on Mon, 08 Nov 2010 16:13:00 GMT
View Forum Message <> Reply to Message

Hello,

Kenneth P. Bowman wrote:
> In article
> <8c77484b-ac8e-4579-ab69-347ff498411f@k14g2000pre.googlegroups.com>,
> Paul <paulstaten@gmail.com> wrote:
>

>> Now that -1 is a valid array index, does it make sense for WHERE to
>> return it as a null result?
>
> If you are not in the habit of always checking the 'count'
> returned by WHERE, now is the time to start. :-)
>
> Oh, and update your legacy code also.

Please permit me to reiterate/second/etc what Ken said above, as well as to apologise in advance for the preachy nature of the following comments:

You should always program defensively - do not rely on a peculiar behaviour, e.g. in this case WHERE returning -1 meaning "something went wrong". Checking for the "count" result returned by WHERE not only avoids the issue you raised, it also makes the code self-documenting. That is, the following

```
  idx = WHERE( xArray EQ yScalar )
  IF ( idx EQ -1 ) THEN ....
```

is really only understandable to the more experienced IDLer, whereas

```
  idx = WHERE( xArray EQ yScalar, count=n_matches )
  IF ( n_matches EQ 0 ) THEN ....
```

is much clearer. One should assume someone else will eventually be maintaining the code you write today. Be kind to them.

And, since I'm on a bit of proselytising roll right now, let me quote from James O. Coplien's forward in "Clean Code", by Robert C. Martin:

"The French poet Paul Valery advises us that a poem is never done and bears continual rework, and to stop working on it is abandonment. .... We abandon our code early, not because it is done, but because our value system focuses more on outward appearance than on the substance of what we deliver. This inattentiveness costs us in the end: A bad penny always shows up. Research, neither in industry nor in academia, humbles itself to the lowly station of keeping code clean. .... Quality is the result of a million selfless acts of care -- not just of any great method that descends from the heavens."

Yea, verily, here endeth the lesson. (Again, my apologies :o)

cheers,

paulv