
Subject: comparing numbers using WHERE doesn't work
Posted by [Miklos Z. Kiss](#) on Sat, 02 Jun 2001 19:15:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi folks:

I have noticed a strange occurrence when I try to compare a number with all the elements of an array. I'll show my situation in an example.

Consider the following array:

a = [2.0000, 2.1000, 2.2000, 2.3000, 2.4000, 2.5000, 2.6000].

Let, x = 2.2000, y = 2.4000

So, I want to find which elements of a match up with x and y.
Naturally, I do the following:

```
index0 = WHERE(a EQ x)
index0 = index[0] ; convert from 1-D array to single integer.
index1 = WHERE(a EQ y)
index1 = index[0]
```

So, when I execute, I may get the following:

```
PRINT, index0
```

```
2
```

```
PRINT, index1
```

```
-1
```

The output clearly tells me that x occurs at a[2] and that y does not exist in a at all. By inspection, IDL should return the value 4. Does anyone have any idea why this happens? Thanks,

Mik Kiss

Subject: Re: comparing numbers using WHERE doesn't work
Posted by [John-David T. Smith](#) on Mon, 04 Jun 2001 21:24:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Miklos Z. Kiss" wrote:

>

> Hi folks:

> I have noticed a strange occurrence when I try to compare a number with
> all the elements of an array. I'll show my situation in an example.

> Consider the following array:

> a = [2.0000, 2.1000, 2.2000, 2.3000, 2.4000, 2.5000, 2.6000].

> Let, x = 2.2000, y = 2.4000

> So, I want to find which elements of a match up with x and y.

> Naturally, I do the following:

>
> index0 = WHERE(a EQ x)
> index0 = index[0] ; convert from 1-D array to single integer.
> index1 = WHERE(a EQ y)
> index1 = index[0]

What is the vector "index" you reference? I think you must have meant

index0=index0[0]

JD

Subject: Re: comparing numbers using WHERE doesn't work
Posted by [Martin Schultz](#) on Tue, 05 Jun 2001 13:18:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Miklos Z. Kiss" <mzkiss@unity.ncsu.edu> writes:

> Hi folks:
> I have noticed a strange occurrence when I try to compare a number with
> all the elements of an array. I'll show my situation in an example.
> Consider the following array:
> a = .
> Let, x = 2.2000, y = 2.4000
> So, I want to find which elements of a match up with x and y.
> Naturally, I do the following:
>
>
> index0 = WHERE(a EQ x)
> index0 = index ; convert from 1-D array to single integer.
> index1 = WHERE(a EQ y)
> index1 = index
>
>
> So, when I execute, I may get the following:
> PRINT, index0
> 2
>
> PRINT, index1
> -1
>
> The output clearly tells me that x occurs at a and that y does not exist
> in a at all. By inspection, IDL should return the value 4. Does anyone
> have any idea why this happens? Thanks,
>
> Mik Kiss

You must have a strange computer system ;-). The above should normally work, if (1) you are positively sure that all number are single precision or all numbers are double, (2) you didn't get 2.4000 as a result of some computation but explicitly wrote it into the vector.

But these 2 conditions show you exactly why you shouldn't use an EQ comparison on floating point numbers.

Cheers,

Martin

```
--
[ Dr. Martin Schultz  Max-Planck-Institut fuer Meteorologie  [[
[ Bundesstr. 55, 20146 Hamburg  [[
[ phone: +49 40 41173-308  [[
[ fax: +49 40 41173-298  [[
[ martin.schultz@dkrz.de  [[
[ ]]
```

Subject: Re: comparing numbers using WHERE doesn't work
Posted by [R.Bauer](#) on Wed, 06 Jun 2001 11:47:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Miklos Z. Kiss" wrote:

```
>
> Hi folks:
> I have noticed a strange occurrence when I try to compare a number with
> all the elements of an array. I'll show my situation in an example.
> Consider the following array:
> a = [2.0000, 2.1000, 2.2000, 2.3000, 2.4000, 2.5000, 2.6000].
> Let, x = 2.2000, y = 2.4000
> So, I want to find which elements of a match up with x and y.
> Naturally, I do the following:
>
> index0 = WHERE(a EQ x)
> index0 = index[0] ; convert from 1-D array to single integer.
index0 is now redefined by index[0].

> index1 = WHERE(a EQ y)
> index1 = index[0]
index1 is now redefined by index[0].
```

But where is index redefined normally index0 and

index1 should have the same value.

I believe you missed something in your example.

```
>  
> So, when I execute, I may get the following:  
> PRINT, index0  
> 2  
>  
> PRINT, index1  
> -1  
>
```

Reimar

--

Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)

Forschungszentrum Juelich

email: R.Bauer@fz-juelich.de

<http://www.fz-juelich.de/icg/icg1/>

=====

a IDL library at Forschungszentrum Juelich

http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.html

<http://www.fz-juelich.de/zb/text/publikation/juel3786.html>
