Subject: Re: array handling
Posted by David Fanning on Sat, 06 Jan 2007 15:01:46 GMT
View Forum Message <> Reply to Message

jochem.verelst@gmail.com writes:

> I am a newbie in IDL. Although I cannot handle the following problem, I
> am sure for the 'Masters' this is peanuts.
>
> The problem:
> I wish to filter an array of redundant data in one line. This is a part
> of the line to be filtered:
>
> -Inf -NaN -Inf 0.542373 2.39394 2.28125
> 0.000000 0.000000 0.000000
> 3.21053 1.21839 0.000000 -NaN 0.967213
> 2.03448 0.724138 0.000000 0.000000 0.000000
> -NaN
>
> With WHERE I can get rid of the zeros, but I dont know how to exclude
> the '-NaN' and '-Inf'.
>
> Any help would be most welcome,

Here is an article that will help:

http://www.dfanning.com/tips/check_nan.html

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: array handling
Posted by Brian Larsen on Sat, 06 Jan 2007 16:15:09 GMT
View Forum Message <> Reply to Message

Something I have always wondered is the difference between using
finite() and just using ne?

As an example:
IDL> array = [1.0, 2.0, !Values.F_NAN, 4.0, !Values.F_NAN]
IDL> Print, Where(NOT Float(Finite(array)))

```
         2           4
IDL> print, where(array ne array)
         2           4
```

What is the fundamental difference here? Speed? Style? or something
else entirely?

Cheers,

Brian

On Jan 6, 8:01 am, David Fanning <n...@dfanning.com> wrote:
> jochem.vere...@gmail.com writes:
>> I am a newbie in IDL. Although I cannot handle the following problem, I
>> am sure for the 'Masters' this is peanuts.
>
>> The problem:
>> I wish to filter an array of redundant data in one line. This is a part
>> of the line to be filtered:
>
>> -Inf   -NaN   -Inf   0.542373   2.39394   2.28125
>>   0.000000   0.000000   0.000000
>>   3.21053   1.21839   0.000000   -NaN   0.967213
>> 2.03448   0.724138   0.000000   0.000000   0.000000
>> -NaN
>
>> With WHERE I can get rid of the zeros, but I dont know how to exclude
>> the '-NaN' and '-Inf'.
>
>> Any help would be most welcome,Here is an article that will help:
>
>   http://www.dfanning.com/tips/check_nan.html
>
> 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: array handling
Posted by Michael Galloy on Sat, 06 Jan 2007 16:39:13 GMT
View Forum Message <> Reply to Message

Brian Larsen wrote:
> Something I have always wondered is the difference between using
> finite() and just using ne?
>
> As an example:
> IDL> array = [1.0, 2.0, !Values.F_NAN, 4.0, !Values.F_NAN]
> IDL> Print, Where(NOT Float(Finite(array)))
> 2 4
> IDL> print, where(array ne array)
> 2 4
>
> What is the fundamental difference here? Speed? Style? or something
> else entirely?

One thing is that FINITE can handle NaN's and infinity all in one step.

By the way, it took me a second to figure out why you were converting to
float in the FINITE example. Maybe

IDL> print, where(~finite(array))

would be better since FINITE returns only 0's and 1's (as long as you
have IDL 6.0+)?

Mike
--
www.michaelgalloy.com

Subject: Re: array handling
Posted by David Fanning on Sat, 06 Jan 2007 16:47:33 GMT
View Forum Message <> Reply to Message

Brian Larsen writes:

> Something I have always wondered is the difference between using
> finite() and just using ne?
>
> As an example:
> IDL> array = [1.0, 2.0, !Values.F_NAN, 4.0, !Values.F_NAN]
> IDL> Print, Where(NOT Float(Finite(array)))
> 2 4
> IDL> print, where(array ne array)
> 2 4

>
> What is the fundamental difference here? Speed? Style? or something
> else entirely?

I don't know the answer, but I would guess the latter formulation
would lead some programmers to a general feeling of unease
regarding treating things that aren't numbers as if they were.
There are no hard and fast rules for such things, I don't think,
so you are at the mercy of programmers you don't know and
implementations that could, I guess, change in the future.

Plus, you get that damn "Arithmetic error: Floating illegal operation"
message when you try it. That generally causes a loss of confidence
among customers. :-(

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: array handling
Posted by David Fanning on Sat, 06 Jan 2007 16:51:30 GMT
View Forum Message <> Reply to Message

Michael Galloy writes:

> By the way, it took me a second to figure out why you were converting to
> float in the FINITE example. Maybe
>
> IDL> print, where(~finite(array))
>
> would be better since FINITE returns only 0's and 1's (as long as you
> have IDL 6.0+)?

That confused me, too. I had to go read the damn article myself. :-)

But I did update the page to this more compact formulation
while I was doing it.

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: array handling
Posted by news.qwest.net on Mon, 08 Jan 2007 17:43:34 GMT
View Forum Message <> Reply to Message

"Brian Larsen" <balarsen@gmail.com> wrote in message
news:1168100109.828745.131590@11g2000cwr.googlegroups.com...
> Something I have always wondered is the difference between using
> finite() and just using ne?
>
> As an example:
> IDL> array = [1.0, 2.0, !Values.F_NAN, 4.0, !Values.F_NAN]
> IDL> Print, Where(NOT Float(Finite(array)))
> 2 4
> IDL> print, where(array ne array)
> 2 4
>
> What is the fundamental difference here? Speed? Style? or something
> else entirely?
>
> Cheers,
>
> Brian

while where(array ne array) is clever, short, and possibly fast
one should note (from IDL help):
"On Windows, using relational operators such as EQ and NE
with the values infinity or NaN (Not a Number) causes an "illegal operand"
error"

Programmers will generally avoid depending on a result that flags an error,
and is not strictly defined in the language. The results may change in
different versions
of IDL, and perahps on different operating systems. Who knows, in version 7,
IDL may
decide to make nan eq nan return 1.

Cheers,
bob

Subject: Re: array handling
Posted by Brian Larsen on Thu, 11 Jan 2007 15:27:15 GMT
View Forum Message <> Reply to Message

>> Brian while where(array ne array) is clever, short, and possibly fast
> one should note (from IDL help):
> "On Windows, using relational operators such as EQ and NE
> with the values infinity or NaN (Not a Number) causes an "illegal operand"
> error"

This would explain why I had never seen the error that David mentioned before. In Linux it just goes on its merry way w/o errors. But in the future I think I will move to finite.

Brian