Subject: n_elements and NaN
Posted by R.Bauer on Tue, 25 Jan 2011 20:26:42 GMT
View Forum Message <> Reply to Message

Just recognized some fun with NaN or more arguments against NaN.

n_elements haven't a NaN keyword

Reimar

Subject: Re: n_elements and NaN
Posted by rogass on Tue, 25 Jan 2011 22:19:03 GMT
View Forum Message <> Reply to Message

On 25 Jan., 21:26, Reimar Bauer <R.Ba...@fz-juelich.de> wrote:
> Just recognized some fun with NaN or more arguments against NaN.
>
> n_elements haven't a NaN keyword
>
> Reimar

Yes,
a nan check for all routines would be wonderful...

Schade an sich

CR

Subject: Re: n_elements and NaN
Posted by lecacheux.alain on Wed, 26 Jan 2011 08:15:38 GMT
View Forum Message <> Reply to Message

On 25 jan, 21:26, Reimar Bauer <R.Ba...@fz-juelich.de> wrote:
> Just recognized some fun with NaN or more arguments against NaN.
>
> n_elements haven't a NaN keyword
>
> Reimar

Why should it do ?
"N_elements(x)" is the old way for "(Size(x))[-1]" or "Size(x,/
N_ELEMENTS)".
What you want is "where(finite(x), COUNT=number_of_finite_elements,
NCOMP=number_of_nan)".

The two statements adress two different things: the x size and x
content.
alx.

Subject: Re: n_elements and NaN
Posted by R.Bauer on Wed, 26 Jan 2011 17:20:04 GMT

View Forum Message <> Reply to Message

Am 26.01.2011 09:15, schrieb alx:
> On 25 jan, 21:26, Reimar Bauer <R.Ba...@fz-juelich.de> wrote:
>> Just recognized some fun with NaN or more arguments against NaN.
>>
>> n_elements haven't a NaN keyword
>>
>> Reimar
>
> Why should it do ?
> "N_elements(x)" is the old way for "(Size(x))[-1]" or "Size(x,/
> N_ELEMENTS)".
> What you want is "where(finite(x), COUNT=number_of_finite_elements,
> NCOMP=number_of_nan)".
> The two statements adress two different things: the x size and x
> content.
> alx.

This is only a workaround. Until NaN is not completly supported we will always have risks to use it. Or having more complicated code as usually needed. If you for example expect only to have Long values you would never expect NaN numbers there and the data of type float.

Reimar

Subject: Re: n_elements and NaN
Posted by Jeremy Bailin on Thu, 27 Jan 2011 15:24:29 GMT

View Forum Message <> Reply to Message

On Jan 26, 12:20 pm, Reimar Bauer <R.Ba...@fz-juelich.de> wrote:
> Am 26.01.2011 09:15, schrieb alx:
>
>> On 25 jan, 21:26, Reimar Bauer <R.Ba...@fz-juelich.de> wrote:
>>> Just recognized some fun with NaN or more arguments against NaN.
>

>>> n_elements haven't a NaN keyword
>
>>> Reimar
>
>> Why should it do ?
>> "N_elements(x)" is the old way for "(Size(x))[-1]" or "Size(x,/
>> N_ELEMENTS)".
>> What you want is "where(finite(x), COUNT=number_of_finite_elements,
>> NCOMP=number_of_nan)".
>> The two statements adress two different things: the x size and x
>> content.
>> alx.
>
> This is only a workaround. Until NaN is not completly supported we will
> always have risks to use it. Or having more complicated code as usually
> needed. If you for example expect only to have Long values you would
> never expect NaN numbers there and the data of type float.
>
> Reimar

I'm with alx on this - I would be very annoyed if n_elements looked at
the contents of what it's given rather than just the syntactic
structure (it would be *much* slower, for one thing!).

-Jeremy.