Subject: Re: Testing for NODATA presence in a dataset
Posted by tam on Mon, 30 Dec 2002 14:52:21 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Tom McGlynn (tam@lheapop.gsfc.nasa.gov) writes and
> Bill Thompson confirms:
>
>
>> That doesn't distinguish NaN from the infinities.
>> The standard trick in any language for looking for NaN's is
>>
>> if x ne x then begin
>> print,'This is a NaN'
>> endif else ...
>
>
> Humm, well, consider this little test in IDL 5.5 or 5.6
> for Windows:
>
> IDL> a = [1.0, 2.0, !Values.F_NAN, 4.0, !Values.F_NAN]
> IDL> print, a
> 1.00000 2.00000 NaN 4.00000 NaN
> IDL> print, a(1)
> 2.00000
>
> All well and good so far. Test the algorithm.
>
> IDL> if a(1) ne a(1) THEN print, 'NAN' ELSE print, 'Number'
> Number
>
> Perfect. Working fine. Now text NAN.
>
> IDL> print, a(2)
> NaN
> IDL> if a(2) ne a(2) THEN print, 'NAN' ELSE print, 'Number'
> Number
> % Program caused arithmetic error: Floating illegal operand
>
> Oh, oh. What's up with that? And a floating illegal operand to
> boot. :-(
>
> How about the array in general?
>
> IDL> print, array ne array
> 0 0 0 0 0
> % Program caused arithmetic error: Floating illegal operand
>

> Humm. I presume you guys have a reason for thinking
> like you do. Any insights?
>
> Cheers,
>
> David

Just to follow up on Bill's message.... I did warn in my first message that interpreters had been known to screw up this comparison, but I believe the behaviour you see above is clearly non-compliant with the IEEE 754 floating point standard. I almost never run IDL under Windows, but I'd call this a bug -- though I daresay RSI will call it a feature.

Using IDL 5.2 under Linux I have:

```
IDL> a=sqrt(-1)
%Program caused arithmetic error.  Floating illegal operand.
IDL> print, a
         -NaN
IDL> print a ne a
   1
IDL> z=[0,0,a,a,0]
IDL> print, z ne z
   0   0   1   1   0
```

I believe this to be 'correct' behavior but it appears that it is not universally implemented this way within IDL. Of course IDL has been implemented on non-IEEE machines (e.g., VAX) and so completely consistent behavior may be impossible.

Let me add my apologies though for misleading anyone looking for how to actually do something, rather than how they should be able to do it.

Regards,
Tom McGlynn