Subject: Re: Testing for NODATA presence in a dataset
Posted by tam on Mon, 23 Dec 2002 02:33:44 GMT
View Forum Message <> Reply to Message

David Fanning <david@dfanning.com> wrote in message
news:<MPG.186e97123b2778b2989a81@news.frii.com>...
> Jonathan Greenberg (greenberg@ucdavis.edu) writes:
>
>> I'm having a problem testing for whether an entry in an array is NAN --
>> doing something like:
>>
>> If (value EQ !VALUES.F_NAN) then begin
>> print,'Not a number'
>> Endif else begin
>> print,'Is a number!'
>> Endelse
>>
>> Will always return 'Is a number', even if I set:
>> value = !VALUES.F_NAN
>>
>> What's going wrong with this?
>
> The problem is that NAN is ... well, not a number.
> Thus, you can't use it in expressions that
> require a number. (Think of it as a mathematical
> Catch-22, if you like.)
>
> The proper way to write this code is like this:
>
> If Finite(value) EQ 0 then begin
> print,'Not a number'
> Endif else begin
> print,'Is a number!'
> Endelse

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 ...
```

This can get optimized away if the compiler/interpreter
is poorly designed. Seemed to work for me in a quick
test though for IDL 5.2 on Linux. NaN's are not equal
to anything --- even themselves.

Regards,
Tom McGlynn