Subject: Incorrect behavior of /NAN
Posted by K. Bowman on Thu, 04 Apr 2002 15:40:11 GMT
View Forum Message <> Reply to Message

I tried to incite some interest in this last week, but had no takers,
so I'll try again. ;-)

I believe that the behavior of the TOTAL function is incorrect in
the case where all of the data are NaN and the /NAN keyword is set.

Here are several possible cases:

```
IDL> a = !values.f_nan
IDL> print, total([a, 1.0])
         NaN
IDL> print, total([a, 1.0], /nan)
     1.00000
IDL> print, total([a, a])
         NaN
IDL> print, total([a, a], /nan)
     0.00000
```

I believe that the last case is incorrect.

The documentation for /NAN says "Elements with the value NaN are treated
as missing data." In the last case there are no valid data, so how can
their sum be zero?

I think this is probably a simple error in the TOTAL algorithm.
Rather than computing the "sum of all the non-NaN values", it is
probably doing "set sum to zero and then add all non-Nan values".

For comparison, the MEAN function does not behave this way.

```
IDL> print, mean([a, 1.0])
         NaN
IDL> print, mean([a, 1.0], /nan)
     1.00000
IDL> print, mean([a, a])
         NaN
IDL> print, mean([a, a], /nan)
         NaN
```

Before I submit something (at least a question) to RSI, does anyone
have comments?

Ken Bowman