In article
<d012ae88-d64f-4ea7-b015-9efdc089a44c@u16g2000pru.googlegroups.com>,
 Edward Hyer <ejhyer@gmail.com> wrote:

>> You can turn off floating-point error messages by setting
>>

>>

>> floating-point exceptions would be more important than having
>> "clean" output.
>
> Many of my routines ruthlessly and systematically divide by zero,
> because the denominator is Ndata.
> Besides, if floating-point errors are so important, how come IDL won't
> tell me where they happened? :)

As David pointed out, you can set

   !EXCEPT = 2

to find out where errors are occurring.

The danger with ignoring floating-point exceptions is that they can
be occurring in multiple places (some benign, some malignant). With
the default (!EXCEPT = 1), you only get one notification. This raises
the possibility of undetected errors.

In my (somewhat obsessive) view, a properly function program should
not generate floating-point errors, except possibly underflows.

It gives me shivers when someone says, "My program is working fine.
Just ignore those floating-point errors."

Ken