
Subject: Re: More on Exp bugs
Posted by [steinhh](#) on Wed, 18 Dec 1996 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Peter Mason wrote:

|> -I realise that this is a MAJOR stir, but I was wondering what people's views
|> -are on IDL's "NaN" and "Infinity" support?
|> -Personally: I haven't yet implemented "Infinity" in my IDL programs,
|> -and I haven't used "NaN" much at all. I like the idea of "NaN", but I
|> -started many of my programs before it was around in IDL, and so I found other
|> -ways to cope with "bad values" and the like.
|>

In article <598caq\$6g9@cpc3.uea.ac.uk>, f055@uea.ac.uk (T.Osborn) writes:

|> NaN is, I think, a good improvement in dealing with missing data values - but
|> perhaps only because the previous support was so patchy. But remaining
|> problems with it are:
|>

A major problem that I see is that only "machines which implement the IEEE standard for binary floating-point arithmetic have two special values for undefined results".

Which platforms do have IEEE FP arithmetic, and which platforms do not?
In other words, there's no way of using NaN values in IDL in any portable sense, is there? If your program uses and relies upon NaN/Infinity, what happens when you run on a platform that doesn't support it?

IMHO, this is a major obstacle for using this (in principle very good) idea. If you have to put in extra lines of code to handle these exceptions on non-IEEE platforms anyway, well, there's not really any point, is there? You'll get *portable* and *shorter* programs devising your own schemes and using only those, although at the cost of efficiency.

I, for one, would prefer that RSI wouldn't waste time implementing new "features" that will not be available on all their platforms. Sure, you may think that some new stuff is cool etc, and that it might be a nice sales pitch. But for the "big" customers, writing software that has to run on as many platforms as possible, there's just no added value at all in such gizmos. These are also the people expanding your list of potential customers (those that want to run the software).

Now, if there was a way to turn on/off "fudged" NaN/Infinity values for *any* platform, that would do the trick. Yes - it would be at the expense of efficiency, but not as much as writing IDL statements to handle the cases.

|> 1) You can't use it to indicate missing values in integer variables (a
|> problem if part of a calculation involves integers, even if the start and end
|> points do not).

Quite agree. There should be a way of specifying NaN_INTEGER, NaN_LONG, (and
perhaps even NaN_BYTE!), as well as NaN_FLOAT/DOUBLE for non-IEEE FP platforms.
Sure, it takes away a little speed, a little of the accessible range of values,
but all of this should be *optional* and *platform independent*.

|>
|> -I can't be bothered with FP underflows (just give me 0).
|>
|> Hear hear.
|>

Quite agree...

Stein Vidar
