Subject: Unknown #INFO

Posted by Luis Oliveira on Wed, 11 Nov 1998 08:00:00 GMT

View Forum Message <> Reply to Message

Hi,

I have received this output from PRINT, when I passed it an array:

```
prf->tprfdata::get(w='velocitycomponents')
   1.#INF0
              1.#INF0
                          1.#INF0
                                     1.#INF0
                                                1.#INF0
           1.#INF0
1.#INF0
                      1.#INF0
   1.#INF0
              1.#INF0
                         1.#INF0
                                    1.#INF0
                                                1.#INF0
1.#INF0
           1.#INF0
  -1.#INF0
              -1.#INF0
                         -1.#INF0
                                    -1.#INF0
                                               -1.#INF0
-1.#INF0
          -1.#INF0
                      -1.#INF0
                         -1.#INF0
  -1.#INF0
              -1.#INF0
                                    -1.#INF0
                                               -1.#INF0
-1.#INF0
           -1.#INF0
              3.80000
                          5.30000
                                     9.80000
   4.40000
                                                8.90000
9.30000
           11.3000
                      4.70000
   6.90000
              15.1000
                         -132.000
                                    -40.9000
                                                -66.7000
-126.800
           -156.700
```

Please, if you have steped already in this "1.#INFO" or "-1#INFO" thing let me know,

Luis

Subject: Re: Unknown #INFO

Posted by davidf on Fri, 13 Nov 1998 08:00:00 GMT

View Forum Message <> Reply to Message

Hi Folks,

Here is a recent e-mail sent to me and posted with his permission from someone who *absolutely* knows what is going on here with this NAN business:

- > Different operating systems use different representations for
- > Nan and Infinity. Some, like Windows or HP-UX are quite bizzare, and not all
- > can be output and then input successfully, even on the platform that produces
- > them. This is all a function of the standard C library (stdio) on the given
- > platform. It has always been thus...

>

- > For IDL 5.2, I pulled the reading and writing of this stuff into IDL
- > code rather than leaving it to the stdio on the target platform. That
- > accounts for the pleasing consistency that you've observed in IDL 5.2Beta.
- > There is a new keyword to PRINT and STRING (STDIO_NON_FINITE) that overrides

- > this and allows the stdio to produce the native output. I don't expect many
- > people to use this keyword, but it's there in case you have some other
- > program on a given platform that understands (for example) "1.#INFO"
- > but not "Infinity".

>

- > I also made IDL on input understand all the various odd platform specific
- > representations for this, so you can read a "1.#INFO" produced on a
- > Windows box on other platforms (like say, Solaris) without any extra effort.

- > A related improvement is that the FINITE function has new keywords
- > ("INFINITY" and "NAN") to make dedecting them in a cross platform way
- > easier.

>

- > There's a section on this stuff in the "What's New" book, but most people
- > have not seen it yet.

Cheers.

David

David Fanning, Ph.D.

Fanning Software Consulting E-Mail: davidf@dfanning.com

Phone: 970-221-0438, Toll-Free Book Orders: 1-888-461-0155 Coyote's Guide to IDL Programming: http://www.dfanning.com/