In article <MPG.10b3c2392717b0898971a@news.frii.com>,
David Fanning <davidf@dfanning.com> wrote:
>
> IDL> Help, !Values, /Structure
> ** Structure !VALUES, 4 tags, length=24:
>   F_INFINITY   FLOAT   1.#INF0
>   F_NAN   FLOAT   1.#QNAN
>   D_INFINITY   DOUBLE   1.#INF000
>   D_NAN   DOUBLE   1.#QNAN00
>
> I don't know. This may be a Windows thing. Looks a little
> Bill Gatesh, doesn't it. :-)

Yes it does look windowish. It probably has something to do
with the compiler they used to make their windows version. What
concerns me is how this affects reading and writing of text files.
In the past I have relied on IDL correctly parsing 'Inf' and 'NaN'
in input files. You can see the obvious portability problem.

I'm not sure if IEEE specifies the text representation of 'Inf'
and 'NaN' or if a program is required to be able to correctly parse
them on input.

Out of curiousity, could you try the following?

print,float('Inf')
print,float('1.#INF0')

(I'm guessing that this will give the same result as read on the same strings.)

My sun gives this:
IDL> print,!version
{ sparc sunos unix 5.0 Apr 28 1997}
IDL> print,!values
{   Inf   NaN   Infinity   NaN}
IDL> print,float('Inf')
        Inf
IDL> print,float('1.#INF0')
     1.00000

Thanks,

Eric

--
Eric Korpela | An object at rest can never be
korpela@ssl.berkeley.edu | stopped.
Click for home page.