Subject: Re: NaN or 0.0
Posted by marq on Wed, 21 May 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Hi Christophe,

you wrote:

```
> Under Linux:
> IDL> print,exp(alog(0))
>         -NaN
> % Program caused arithmetic error: Floating illegal operand
>
> In both case, the program say I'm doing something wrong,
> but with the Unix version, I can continue to work...
>
> Is it possible to say the Linux-IDL not to use this
> &*%$%# NaN absorbing element?
> It seems to me it as something to do with something
> called IEEE, but I don't understand what it is ;-)
```

...it's just a standard how to handle floating point exceptions. On IEEE machines, invalid math operations (like your one) result in an NaN (i.e., Not a Number) or something like 'Infinity', plus a signal delivered to the program, idl in this case. It is then up to the program how to react to the floating point exception - may that be halting (like VMS always does) or just reporting the occurence of the exception, but continuing as if nothing had happened.

With my linux version of IDL (which is 5.0 beta 6), a system variable '!EXCEPT' exists which controls when IDL issues its warnig: never (!EXCEPT = 0), _after_ a function or procedure has finished (!EXCEPT = 1, which seems to be the default), or immediately after the program line that caused the exception (!EXCEPT = 2).

In your case:

```
  IDL> print, !except
        1
  IDL> print, exp(alog(0))
          NaN
  % Program caused arithmetic error: Floating illegal operand
  IDL> !except = 2
  IDL> print, exp(alog(0))
          NaN
  % Program caused arithmetic error: Floating illegal operand
  % Detected at  $MAIN$
```

Hope this helps a bit...

Regards,

Chris.

--- ---------------
Christian Marquardt

Meteorologisches Institut der | tel.: (+49) 30-838-71170
Freien Universitaet Berlin | fax.: (+49) 30-838-71167
Carl-Heinrich-Becker-Weg 6-10 | email: marq@strat01.met.fu-berlin.de
D-12165 Berlin |
--- ---------------