Subject: Re: differences idl5.5 and idl6.6
Posted by R.G. Stockwell on Wed, 19 Feb 2003 18:17:34 GMT
View Forum Message <> Reply to Message

Kenneth Bowman wrote:
> In article <b30ble$dpt7$1@zam602.zam.kfa-juelich.de>,
> Reimar Bauer <R.Bauer@fz-juelich.de> wrote:
>
>
>> on aix the result is a bit more different,
>> IDL> a=1./0 & print,a,fix(a)
>> Inf -1
>>
>>
>> I would prefer as result NaN!
>
>
> The IEEE standard defines distinct special values for Inf (result of
> division by zero) and NaN (result of operations on Inf's and Nan's).
> Being able to distinguish them is probably a good thing. The FINITE
> function has keywords for this purpose. Without keywords, FINITE
> detects both.
>
> Unfortunately, perhaps, standard integer formats do not include
> "special" values to represent inifinties, Nan's, etc. You have to
> handle them yourself. Hence all those ASCII data sets filled with
> -999's!
>
> Conversion of IEEE Inf's and NaN's to integers may well be system
> dependent.
>
> Ken

With integers using two's complement, there is no representation
for these (nan/inf) values (every bit pattern is a valid integer).

The standard way (in C Java IDL etc, perhaps a IEEE standard) to handle
integer division is to flag an exception if the divisor is 0, and
the result is undefined.
It seems that IDL (sensibly) follows the rule of integer math results
in an integer (rather than automatically casting the result to a float, and the
appropriate nan/inf). It is up to the programer to decide if they want
a float division or an integer division.

Also, casting a floating point nan/inf to integer should also throw
an exception (i.e. there is no conversion of inf/nan to integer).

Cheers,
bob