Subject: Re: differences idl5.5 and idl6.6
Posted by Dick Jackson on Wed, 19 Feb 2003 15:54:54 GMT
View Forum Message <> Reply to Message

"Reimar Bauer" <R.Bauer@fz-juelich.de> wrote in message
news:3E539FE4.4000801@fz-juelich.de...
> Hi,
>
> today I found a difference in handling infite numbers by the new
version.
>
>
> IDL 5.6
>
> IDL> a=1./0 &print,a,fix(a)
> Inf -32768
>
>
> IDL 5.5
>
> IDL> a=1./0 &print,a,fix(a)
> Inf 0
>
>
>
> So I can't say which is right both results are terrible.
>
> Now it is totally clear that this case must be tested.

I don't see any change in behaviour on Windows 2000 Pro; in 5.4, 5.5 and
5.6, I get these identical results:

```
IDL> a=1./0 &print,a,fix(a)
       Inf       0
% Program caused arithmetic error: Floating divide by 0
% Program caused arithmetic error: Floating illegal operand
```

Cheers,
--
-Dick

Dick Jackson / dick@d-jackson.com
D-Jackson Software Consulting / http://www.d-jackson.com
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392

Subject: Re: differences idl5.5 and idl6.6

Dick Jackson wrote:
> "Reimar Bauer" <R.Bauer@fz-juelich.de> wrote in message
> news:3E539FE4.4000801@fz-juelich.de...
>
>> Hi,
>>
>> today I found a difference in handling infite numbers by the new
>
> version.
>
>>
>> IDL 5.6
>>
>> IDL> a=1./0 &print,a,fix(a)
>> Inf -32768
>>
>>
>> IDL 5.5
>>
>> IDL> a=1./0 &print,a,fix(a)
>> Inf 0
>>
>>
>>
>> So I can't say which is right both results are terrible.
>>
>> Now it is totally clear that this case must be tested.
>
>
> I don't see any change in behaviour on Windows 2000 Pro; in 5.4, 5.5 and
> 5.6, I get these identical results:
>
> IDL> a=1./0 &print,a,fix(a)
> Inf 0
> % Program caused arithmetic error: Floating divide by 0
> % Program caused arithmetic error: Floating illegal operand
>
> Cheers,
> --
> -Dick
>
> Dick Jackson / dick@d-jackson.com
> D-Jackson Software Consulting / http://www.d-jackson.com
> Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392
>
>

Thanks Dick,

then it seems to be a bug on linux and aix

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!

regrads

Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
 -- -------
        a IDL library at ForschungsZentrum Juelich
  http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html
 == =======

Subject: Re: differences idl5.5 and idl5.6
Posted by Paul Van Delst[1] on Wed, 19 Feb 2003 16:56:20 GMT
View Forum Message <> Reply to Message

Reimar Bauer wrote:

>
> Dick Jackson wrote:
>> "Reimar Bauer" <R.Bauer@fz-juelich.de> wrote in message
>> news:3E539FE4.4000801@fz-juelich.de...
>>
>>> Hi,
>>>
>>> today I found a difference in handling infite numbers by the new
>>
>> version.
>>
>>>
>>> IDL 5.6
>>>
>>> IDL> a=1./0 &print,a,fix(a)
>>>   Inf -32768
>>>
>>>
>>> IDL 5.5
>>>
>>> IDL> a=1./0 &print,a,fix(a)
>>>   Inf   0
>>>
>>>
>>>
>>> So I can't say which is right both results are terrible.
>>>
>>> Now it is totally clear that this case must be tested.
>>
>>
>> I don't see any change in behaviour on Windows 2000 Pro; in 5.4, 5.5 and
>> 5.6, I get these identical results:
>>
>> IDL> a=1./0 &print,a,fix(a)
>>   Inf   0
>> % Program caused arithmetic error: Floating divide by 0
>> % Program caused arithmetic error: Floating illegal operand
>>
>> Cheers,
>> --
>> -Dick
>>
>> Dick Jackson   /   dick@d-jackson.com
>> D-Jackson Software Consulting /   http://www.d-jackson.com
>> Calgary, Alberta, Canada   / +1-403-242-7398 / Fax: 241-7392
>>
>>
>

> Thanks Dick,
>
> then it seems to be a bug on linux and aix
>
> 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!

Hmm. Wouldn't the integer form of infinity still be infinite? If you think the value of some variable may be infinite, how come you don't test the value _before_ using it in an intrinsic function (e.g. FIX()) ?

I.e.

IF (FINITE(a)) THEN a = FIX(a) ELSE a = !VALUES.F_NAN

And from the IDL docs, it would appear that there is no such thing as an integer infinity or nan (in IDL at least).

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC
Ph: (301)763-8000 x7274
Fax:(301)763-8545

Subject: Re: differences idl5.5 and idl5.6
Posted by K. Bowman on Wed, 19 Feb 2003 17:26:37 GMT
View Forum Message <> Reply to Message

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

Subject: Re: differences idl5.5 and idl5.6
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

Subject: Re: differences idl5.5 and idl5.6
Posted by R.Bauer on Thu, 20 Feb 2003 09:01:55 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:
> Reimar Bauer wrote:
>
>> Dick Jackson wrote:
>>
>>> "Reimar Bauer" <R.Bauer@fz-juelich.de> wrote in message
>>> news:3E539FE4.4000801@fz-juelich.de...
>>>
>>>
>>>> Hi,
>>>>
>>>> today I found a difference in handling infite numbers by the new
>>>
>>> version.
>>>
>>>
>>>> IDL 5.6
>>>>
>>>> IDL> a=1./0 &print,a,fix(a)
>>>>   Inf -32768
>>>>
>>>>
>>>> IDL 5.5
>>>>
>>>> IDL> a=1./0 &print,a,fix(a)
>>>>   Inf   0

>>>>
>>>>
>>>>
>>>> So I can't say which is right both results are terrible.
>>>>
>>>> Now it is totally clear that this case must be tested.
>>>
>>>
>>> I don't see any change in behaviour on Windows 2000 Pro; in 5.4, 5.5 and
>>> 5.6, I get these identical results:
>>>
>>> IDL> a=1./0 &print,a,fix(a)
>>>   Inf   0
>>> % Program caused arithmetic error: Floating divide by 0
>>> % Program caused arithmetic error: Floating illegal operand
>>>
>>> Cheers,
>>> --
>>> -Dick
>>>
>>> Dick Jackson   /   dick@d-jackson.com
>>> D-Jackson Software Consulting /   http://www.d-jackson.com
>>> Calgary, Alberta, Canada   / +1-403-242-7398 / Fax: 241-7392
>>>
>>>
>>
>> Thanks Dick,
>>
>> then it seems to be a bug on linux and aix
>>
>> 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!
>
>
> Hmm. Wouldn't the integer form of infinity still be infinite? If you think the value of
> some variable may be infinite, how come you don't test the value _before_ using it in an
> intrinsic function (e.g. FIX()) ?

Normally we check this but sometimes we missed it. The problem now is
that it is not easy to find because of the different results it gives on
different platforms.

The routine which was going wrong now was written in 1997. As it was
programmed no one thougt about that it makes sense to use this routine

with only one color.
And for the people which were using it with only one color it works
perfectly in the past.

For crossplatform programming it is better that failures are described
same on each of the idl platforms. (My feeling) In the past it was this way.

regards

Reimar

>
> I.e.
>
> IF (FINITE(a)) THEN a = FIX(a) ELSE a = !VALUES.F_NAN
>
> And from the IDL docs, it would appear that there is no such thing as an integer infinity
> or nan (in IDL at least).
>
> paulv
>

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
---------- -------
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html
========== =======