

---

Subject: bizarre number transformation

Posted by [merlecorp](#) on Thu, 25 Jul 2002 11:46:35 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hello,

I ran into a number transformation error yesterday that is still confusing me this morning. At first I thought I was doing something silly with String() & StrTrim(), but then I wrote a little program (see huh.pro) with no conversions that still contained the problem. FYI, I'm using IDL Version 5.5 Win32 (x86).

The problem is that the number 443496.984 is being turned into the number 443496.969 from basic assignments using Float() or Double(), despite the fact that even floats should easily be able to handle a number this large (floats can handle  $\approx 10^{38}$ , with approximately six or seven decimal places of significance).

Since I knew that I had successfully read in numbers much greater than 443496.984 in the past, I created temp.dat with just the number 443496.984 in it, and read this into a variable, x3. If x3 is cast as a float, it doesn't work, i.e. the number is 443496.969. But, if x3 is cast as a double, then it contains the correct value. Why isn't a float sufficient (443496.984  $\ll 10^{38}$  and contains only 3 decimal places)? And, why doesn't x2=Double(443496.984) produce the correct result?

Here's the code & the output:

```
; -----
```

```
PRO huh
```

```
; -- print with single precision -> NFG
```

```
X1 = Float(443496.984)
```

```
Print, X1, Format='(X1 = ", (F10.3))'
```

```
; -- print with double precision -> NFG
```

```
X2 = Double(443496.984)
```

```
Print, X2, Format='(X2 = ", (F10.3))'
```

```
; -- read the number from a file & print it out -> works fine
```

```
OpenR, lun, 'temp.dat', /Get_Lun, ERROR = err
```

```
; -- Note: X3 must be cast as a double or else the number becomes
```

```
443496.969
```

```
X3 = Double(0)
```

```
ReadF, lun, X3
Print, X3, Format=('"X3 = ", (F10.3))'
```

```
Free_Lun, lun
```

```
END
```

```
IDL> .COMPILE "D:\IDL\workdir\huh.pro"
```

```
% Compiled module: HUH.
```

```
IDL> huh
```

```
X1 = 443496.969
```

```
X2 = 443496.969
```

```
X3 = 443496.984
```

```
; -----
```

Could someone please explain this to me? Or at least duplicate this error so that I don't think that I'm going crazy?

thanks,  
merle

---

Subject: Re: bizarre number transformation

Posted by [Paul Van Delst\[1\]](#) on Fri, 26 Jul 2002 14:39:38 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

James Kuyper wrote:

>

> Craig Markwardt wrote:

>>

>> James Kuyper <kuyper@gscmail.gsfc.nasa.gov> writes:

> ...

>>> The point is, that it's pretty rare to need that many significant

>>> digits. There aren't many real-world numbers that can be measured to

>>> within one part in a billion. Precision needs like that can come up in

>

> ...

> [some examples where such precision is needed]

>> Admittedly those are pretty specialized applications :-)

>

> Precisely. I also do scientific programming, where such needs are pretty

> common. But the vast bulk of the world's programming involves numbers

> that can be represented with adequate accuracy using single precision

> floating point. For instance, how many million-dollar quantities are

> actually measured with a precision of +/- \$1?

I would hope the answer is not any at all. I would think that 10th's of a US dollar/Euro (or even 100th's) would be used. Exchange rates are usually quoted to 1000th's of a US Dollar. This is all a hopeful thought on my part since my past and forecast number of million dollar transactions is, unfortunately, zero.

- > And how many people would
- > actually care if such quantities were in error by \$1 or two?

Depends how many transactions the error occurs in and whether it's a random or systematic error. These little bits can add up to a lot. Which is sort of what this whole thread is about in a way - cumulative numeric precision "errors".

paulv

--

Paul van Delst  
CIMSS @ NOAA/NCEP/EMC      Beer is good.  
Ph: (301)763-8000 x7274      My wife.  
Fax:(301)763-8545

---

---

Subject: Re: bizarre number transformation  
Posted by [mperrin+news](#) on Fri, 26 Jul 2002 17:14:32 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Paul van Delst <paul.vandelst@noaa.gov> wrote:

- > James Kuyper wrote:
- >> Precisely. I also do scientific programming, where such needs are pretty
- >> common. But the vast bulk of the world's programming involves numbers
- >> that can be represented with adequate accuracy using single precision
- >> floating point. For instance, how many million-dollar quantities are
- >> actually measured with a precision of +/- \$1?
- >
- > I would hope the answer is not any at all. I would think that 10th's of a US dollar/Euro
- > (or even 100th's) would be used. Exchange rates are usually quoted to 1000th's of a US
- > Dollar. This is all a hopeful thought on my part since my past and forecast number of
- > million dollar transactions is, unfortunately, zero.

Some brief experimentation with Microsoft Excel (I know, not the world's paragon of financial software, but it's what I have on hand.) seem to indicate that it uses doubles for storing constants entered into cells. It happily handles up to 15 decimal digits without trouble and truncates digits entered after that (not rounds). This allows you to handle up to ten trillion dollars retaining down to the penny - good enough for most purposes indeed.

Quicken on the other hand restricts amounts entered to  
|n| < 9,999,999.99  
and doesn't allow precision greater than 0.01 to be entered at all.

This is either a limitation imposed by the UI which doesn't reflect the limitations of their internal representation, or they're using fixed-width BCD in units of one cent - which would strike me as the Correct Thing to do in this case.

My gut instinct is that most financial software out there probably uses either doubles or BCD. Roundoff errors could easily become a major financial problem if you're Visa.

- Marshall

---

Subject: Re: bizarre number transformation  
Posted by [David Fanning](#) on Fri, 26 Jul 2002 17:33:59 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Marshall Perrin (mperrin+news@arkham.berkeley.edu) writes:

> My gut instinct is that most financial software out there probably uses  
> either doubles or BCD. Roundoff errors could easily become a major  
> financial problem if you're Visa.

For the hundreds of thousands of financial transactions I process every day for my book, I find it advantageous to round one way if I'm collecting and another way if I'm billing.

Cheers,

David

P.S. Let's just say we decided on the deluxe meal plan for the genius as he heads off to college. :-)

--

David W. Fanning, Ph.D.  
Fanning Software Consulting, Inc.  
Phone: 970-221-0438, E-mail: [david@dfanning.com](mailto:david@dfanning.com)  
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>  
Toll-Free IDL Book Orders: 1-888-461-0155

---

Subject: Re: bizarre number transformation  
Posted by [David Fanning](#) on Fri, 26 Jul 2002 17:36:08 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

David Fanning ([david@dfanning.com](mailto:david@dfanning.com)) writes:

> For the hundreds of thousands of financial transactions  
> I process every day for my book, I find it advantageous  
> to round one way if I'm collecting and another way if I'm  
> billing.

?! Oh, you know what I mean. :-(

David

---

---

Subject: Re: bizarre number transformation  
Posted by [R.Bauer](#) on Mon, 29 Jul 2002 09:04:20 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

James Kuyper wrote:

>  
> Reimar Bauer wrote:  
> ...  
>> thanks for these statements we do need double precision too.  
>> I hope someone of INTEL will read sometimes this discussion,  
>> because they have dropped double precision from their processors.  
>  
> Could you give some details on that? I could imagine INTEL producing  
> chips that don't have double precision, for use in contexts where the  
> extra precision isn't needed. I can't imagine them stopping production  
> of processors that do provide double precision.

My last posting was not carefully enough written.

It is wrong to say "they have dropped double precision from their  
processors"

what's I'd like to say is the performance of double precision looks like  
as it

is only emulated. They have added a lot of more features to add speed to  
several things but not for double precision.

In my eyes it looks like they have most interest in multimedia which  
only need

fastest single precision.

We have done here a lot of comparison of intel and amd processors.  
The result is in simple words. You lose a lot of time if you are using  
the actual Pentium 4 system instead of the actual amd if your  
calculation needs double precision.

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](http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro.html)  
=====

---