Subject: Re: Differences between IDL's floats and Java's floats - a problem
Posted by David Fanning on Thu, 13 Nov 2003 13:05:14 GMT
View Forum Message <> Reply to Message

necr@pml.ac.uk writes:

```
> I am currently porting some IDL code across to java, and I've run into
> a couple of snags with the different ways IDL and Java deal with
> numbers. Most of these I have sorted, with trips to the APIs and a
> fair bit of googling. However, there is one I haven't managed to sort
> out yet.
>
> IDL and Java appear to load floating point numbers from a file in very
> different ways.
>
> In IDL, I can read seperate bytes into memory like so:
>
>   myvar=0b & readu, 10, myvar
>
> (where 10 is the filehandle of the open file which is being read).
> This will result in an unsigned byte being retrieved.
>
> I could read the byte in with java using a DataInputStream like so:
>
>   byte a = input.readUnsignedByte();
>
>
> (where input is my open DataInputStream).
>
> Both these pieces of code would give the same result if run on the
> same byte in a file
>
> To read in a float in idl, I would use:
>
>   readu, 10, floatvar
>
>
> (the floatvar would not previously have been set to anything).
>
> And in java:
>
>   float f = input.readFloat();
>
> However, running these two pieces of code will not result in the same
> float being given out.
>
> My Example
```

>
> I read in four bytes from a file, both in IDL and Java. Both systems
> give me the results 0, 64, 206, 67.
>
> If I run the float code on these same four bytes though, IDL will give
> me 412.50 (the value I want), while Java will give 5.951465E-39 -
> clearly not the number I'm looking for!
>
> I have tested this on both a Sun and a Windows machine, and have
> received the same results.
>
> So, has anyone got any ideas as to why this is happening? And more
> importantly, does anyone know what I can do to get the same float
> value being loaded in Java?
>
> Thankyou in advance for any help you can give me.

This is a byte order problem. I don't know Java, but
consider this IDL experiment:

```
IDL> a=[0b, 64b, 206b, 67b]
IDL> print, float(a,0)
      412.500
IDL> b=[67b,206b,64b,0b]
IDL> print, float(b,0)
 5.95146e-039
```

Java is apparently opposite-ended from whatever
it is you are running on. :-)

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http:/www.dfanning.com/
Phone: 970-221-0438, IDL Book Orders: 1-888-461-0155

Subject: Re: Differences between IDL's floats and Java's floats - a problem
Posted by Neil Crosby on Thu, 13 Nov 2003 13:38:32 GMT

View Forum Message <> Reply to Message

David Fanning wrote:
> necr@pml.ac.uk writes:
>

> This is a byte order problem. I don't know Java, but
> consider this IDL experiment:
>
>
> IDL> a=[0b, 64b, 206b, 67b]
> IDL> print, float(a,0)
>   412.500
> IDL> b=[67b,206b,64b,0b]
> IDL> print, float(b,0)
> 5.95146e-039
>
> Java is apparently opposite-ended from whatever
> it is you are running on. :-)
>
> Cheers,
>
> David

Thanks David.

scurries off to try and get it working...

Subject: Re: Differences between IDL's floats and Java's floats - a problem
Posted by R.G. Stockwell on Thu, 13 Nov 2003 16:42:06 GMT
View Forum Message <> Reply to Message

"Neil Crosby" <necr@pml.ac.uk> wrote in message
news:3eda43e9.0311130409.5ee02334@posting.google.com...
> I am currently porting some IDL code across to java, and I've run into
> a couple of snags with the different ways IDL and Java deal with
> numbers. Most of these I have sorted, with trips to the APIs and a
> fair bit of googling. However, there is one I haven't managed to sort
> out yet.
...
> If I run the float code on these same four bytes though, IDL will give
> me 412.50 (the value I want), while Java will give 5.951465E-39 -
> clearly not the number I'm looking for!
...
> Neil

Hi Neil,
as David said, this looks like a endian problem. Check out
 http://www.ibiblio.org/javafaq/books/javaio/ioexamples/07/in dex.html

and there are some classes that deal with little endian inputs.

Cheers,
bob

Subject: Re: Differences between IDL's floats and Java's floats - a problem
Posted by Nigel Wade on Fri, 14 Nov 2003 09:41:17 GMT
View Forum Message <> Reply to Message

R.G. Stockwell wrote:
> "Neil Crosby" <necr@pml.ac.uk> wrote in message
> news:3eda43e9.0311130409.5ee02334@posting.google.com...
>
>> I am currently porting some IDL code across to java, and I've run into
>> a couple of snags with the different ways IDL and Java deal with
>> numbers. Most of these I have sorted, with trips to the APIs and a
>> fair bit of googling. However, there is one I haven't managed to sort
>> out yet.
>
> ...
>
>> If I run the float code on these same four bytes though, IDL will give
>> me 412.50 (the value I want), while Java will give 5.951465E-39 -
>> clearly not the number I'm looking for!
>
> ...
>
>> Neil
>
>
>
> Hi Neil,
> as David said, this looks like a endian problem. Check out
> http://www.ibiblio.org/javafaq/books/javaio/ioexamples/07/in dex.html
>
> and there are some classes that deal with little endian inputs.
>
>
> Cheers,
> bob
>
>

It is. Java DataStreams read in network byte order (big endian).
To read little endian data I read it into a byte[] array and then wrap it in
a ByteBuffer set to ByteOrder.LITTLE_ENDIAN. There's probably a thousand
other ways to to it, I just find that the most straight forward.

--
Nigel Wade, System Administrator, Space Plasma Physics Group,
University of Leicester, Leicester, LE1 7RH, UK
E-mail : nmw@ion.le.ac.uk
Phone : +44 (0)116 2523548, Fax : +44 (0)116 2523555