Subject: Double formatting bug?
Posted by zsolt on Sun, 19 Nov 2006 21:48:40 GMT
View Forum Message <> Reply to Message

I'm using "%0nl" formatting. This is formatting double values as fixed point (integer) number in most cases. But with very large or small numbers (e.g. 1e-16) it shows the floating point number (e.g. 1e-16). This is what I don't need.

I have checked the formatting function:

```
String FormatDouble(double d, int digits, int flags, int pad_exp)
{
 if(IsNull(d))
  return Null;

 double ad = fabs(d);
 bool is_exp = (flags & FD_EXP);
 if(!(flags & FD_FIX))
 {
  is_exp = ad && (ad <= 1e-15 || ad >= 1e15);
  if(flags & FD_REL)
  {
   double bd = ipow10(2 * digits);
   if(ad && (ad * bd <= 1 || ad >= bd))
    is_exp = true;
  }
 }
 if(is_exp)
  return FormatDoubleExp(d, digits, flags, pad_exp);
 else
  return FormatDoubleFix(d, digits, flags);
}
```

The problem is the "if(!(flags & FD_FIX))" block. I searched for FD_FIX in upp sources and I was unable to find any other references to it. Is something missing from parser or is this "if" block absolutely unneeded?
Or how can I enforce fixed point language based formatting?

Subject: Re: Double formatting bug?
Posted by rylek on Mon, 20 Nov 2006 09:12:54 GMT
View Forum Message <> Reply to Message

Hello!

I think that is not a bug. FD_FIX, just as FD_EXP, are bit-coded option flags for the FormatDouble function affecting the choice of number output format. When you use, e.g.,

FormatDouble(32.5, 3, FD_FIX)

it always gets formatted in the "traditional" way (as 32.5). On the other hand, calling

FormatDouble(32.5, 3, FD_EXP)

causes the number to be always formatted in the scientific exponential notation (3.25e1). If neither of the option flags are used, FormatDouble selects the appropriate format automatically based on magnitude of the number being formatted (the code within the if-block).

I believe that these three variants (traditional output notation, exponential output notation, autoselect based on magnitude) correspond to the NFormat specifiers %nf, %ne, and %n, respectively. However the formatting function (RealFormatter) directly calls the appropriate formatting functions as necessary (FormatDoubleExp / FormatDoubleFix) so that the call to FormatDouble is used only exactly to invoke the magnitude-based format autoselection mechanism, so neither FD_FIX nor FD_EXP is used there.

Regards

Tomas

Subject: Re: Double formatting bug?
Posted by zsolt on Mon, 20 Nov 2006 09:58:59 GMT
View Forum Message <> Reply to Message

OK, undersand.
My real problem is, that I want to format double values without decimals and with thousand separator, but always in fixed point format.
Is it possible with formatting patterns, or should I create my own Convert class?

Subject: Re: Double formatting bug?
Posted by rylek on Wed, 22 Nov 2006 12:33:55 GMT
View Forum Message <> Reply to Message

I see. Unfortunately this is currently not possible; the only available formatting variants are ne, nf, and nl, where ne and nf do not generate the thousand separator and nl selects output notation automatically. According to the currently adopted notation, it seems to me cleanest to define another formatting specifiers, nle and nlf, which will format the output using language-dependent behaviour together with notation specification.

Regards

Tomas

Subject: Re: Double formatting bug?
Posted by zsolt on Wed, 22 Nov 2006 14:59:47 GMT
View Forum Message <> Reply to Message

rylek wrote on Wed, 22 November 2006 13:33According to the currently adopted notation, it seems to me cleanest to define another formatting specifiers, nle and nlf, which will format the output using language-dependent behaviour together with notation specification.

Good idea, I agree with you.