
Subject: Re: Value with type float
Posted by [mirek](#) on Sat, 02 Apr 2022 07:43:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

Tom1 wrote on Fri, 01 April 2022 14:03Hi,

The following code portrays the float rounding issue:

```
#include <Core/Core.h>
```

```
using namespace Upp;
```

```
CONSOLE_APP_MAIN
```

```
{  
    Value v(0.002f);  
    Cout() << v << "\n";  
    Cout() << FormatDouble(v) << "\n";  
    Cout() << FormatG(v,7) << "\n";  
}
```

The result of running the above follows:

```
0.0020000000949949
```

```
0.0020000000949949
```

```
0.002
```

```
<--- Finished, press [ENTER] to close the window --->
```

I do not know how to solve this cleanly. In any case a regular user seeing 0.002000000095 in a field where he expects to see 0.002, will not be happy about it. For years I have used roundr() all over the code to fix this up, but having a solution hidden in Core would be awesome!

Best regards,

Tom

OK, so it is correct behaviour. You can simplify that to

```
double x = 0.002;  
DDUMP(x);  
float fx = x;  
x = fx;  
DDUMP(x);
```

which produces exactly the same result. Now for an explanation what is going on here:

0.002 cannot be exactly represented as double. Normal double formatting (as used in DDUMP) rounds for 15 decimal digits which is guaranteed precision and it all yields a "correct" result (in both directions, closes value is chosen and everything is "fine").

Anyway, when you convert it to float, you have to round at equivalent of about 7 valid digits (you have to cut 28 bits of mantissa). In this case, the last mantissa bit of float is rounded up to 1 as that is the closest value to original double. When converting back to double, this cannot be undone, hence you get those "949949" digits at the end.

That said, all of this made me think that if we knew that it is float in Value, we might be able to apply for something like `FormatG(v,7)` instead when displaying it (e.g. in `EditDouble`).

Mirek
