Subject: Re: Value with type float
Posted by mirek on Fri, 03 Jun 2022 08:52:52 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Wed, 11 May 2022 08:20Quote:
Why dont you just use

double hpf;

?
Well, all my signal processing code runs on 32-bit floats, and therefore, the various coefficients/parameters are also floats. It is straight forward to keep it up in the user interface too. If I used doubles in the GUI, I would end up converting forth and back all those parameters. Never wish to go back there, now that I have EditFloat and EditFloatSpin! :)

Anyway, if you feel seriously reluctant to add Null support for float, I can live with it: I figured out a way to do it outside of Core almost cleanly:
constexpr float Nullf = -std::numeric_limits<float>::infinity();
inline bool IsNull(const float& r) { return !(std::abs(r) < std::numeric_limits<float>::infinity()); }
inline void SetNull(float& x) { x = Nullf; }
Still, it would be nicer inside Core... After all, it would introduce only three lines of new code in Core/Defs.h.

Best regards,

Tom

Got a new idea for a bit more universal solution:

The most important difference between Value(double) and Value(float) is the precision used when formatting it. What about to add a general precision hint to Value(double)? That would make possible to e.g. specify arbitrary precision for Json output etc...