
Subject: Re: Value: why not float support?

Posted by [kohait00](#) on Fri, 24 Dec 2010 12:55:12 GMT

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

i admit i have no idea about sql layer in Upp, havent needed it so far. so i dont really understand how this would break it..maybe you can explain it in some 2 phrases..

the new Null value is imho not really a problem.

well, my problem, like described above, is, that i use the cool upp types, which can be packed into Value, as direct feed to the OSC layer, which basicly supports the intrinsic types, but this depends on the spec version, 1.0, or additinoally, 1.1.

1.0 supports int, float, string and blob (binary).

1.1 *additionally* supports int64, double, bool, datetime, etc..

so i decided to simply use the Value as container type, also because there are many implicit conversion possibilities, which spare a lot of work. thats why the extension is crucial, for me at least

i was going well with the 'float as double' handling, as long as keeping spec 1.0 support only, but when, cleanly, extending to 1.1, i need both types to be distinct.

that'd be no problem, i could define and use a custom RichValue<float> and use it just like that, but the conversion mechanisms cant be enriched with user code to support it. IsNumber(), and the implicit converters.

if that'd were possible, no hassle.

but nevertheless, float beeing intrinsic type was finally a motivation to extend it.

i have to admit, char, unsigned types and all that aren't there as well, but they are at least, binary compatible types, means

a char could be banary transported as int, and be interpreted as is later. this holds true for unsigned types as well. but float and double, when transported / serialized over net etc. are *not* binary compatible. though they might be treated equal in cpu, on the net and serialsation, they arent. float is 4 byte, double is 8. this is another pro for float.

if i understood what else might brick with it, i could help thinking of a solution. but i'd rather prefer to make sql layer aware of float than to spare out on float (IMO ofcourse)

afaik, sql also supports both (and some more)

<http://dev.mysql.com/doc/refman/5.0/en/numeric-type-overview.html>

so let me know, where i can help, if so.

EDIT:

i figured out, that all major database concepts (mysql, postgre sql, oracle, mssql) support float (or

single precision).

the only exception is sqlite3, which only supports double, but one could fix this very fast i think, mapping float to double there.
