Subject: Re: Value: why not float support?
Posted by mirek on Sat, 25 Dec 2010 09:48:01 GMT

kohait00 wrote on Fri, 24 December 2010 07:55
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.

AFAIK, there is little difference in promoting char to int and casting back and doing the same with float and double.

Quote:
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.

Sure, and every single SQL engine accepts double there (then perhaps casting it to float internally).

Anyway, that is not a problem. The issue is that existing code, which includes SQL/U++ connectors (there is about 5 of them), does not expect to get something like V_FLOAT as Value.

Currently, any float gets promoted to 'double', so it is send as double and everything works. Make Value to remember that such a type is 'V_FLOAT' and you are sudenly missing appropriate case in switch....

Which is quite easy to fix in connectors and in 'uppsrc', actually, but IMO it is little demanding to expect every U++ users to scan through his code for similar cases.

To do that, I would need some solid reason...

And you still have not provided any clue how do you plan to use this (A bit of end-user code would really be helpful). If you need to support 'char' in your interface, I do not see a single reason why you could not support 'float' the very same way.

Mirek