
Subject: Refactored Value coming....

Posted by [mirek](#) on Sun, 15 Jan 2012 18:02:33 GMT

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

I have spent last 3 weeks refactoring Value, the goal being that most of standard values (like String or int) should be "small value optimized" (in the end, interface allows user-defined types to be svo stored as well).

In the end, I believe that I have succeeded. sizeof(Value) grew to 16, but any standard value except WString, Complex, ValueMap and ArrayMap are now stored 'directly' into those 16 bytes (older implementation always used reference counted shared data).

I have to say I have expected better improvements in terms of speed; what I am getting in syntetic benchmarks is about 40% improvement. But some operations are significantly faster; e.g. storing int to Value is now just two CPU dword assignments (before it involved allocating memory and setting up shared object), likewise extracting int from Value is about 4x faster now.

Plus, memory consumption should be better too.

Means, despite some disappointment, new Value implementation is better - and I have cleaned the interface a bit too. So it will go into mainline sources, but there is no hurry. This is significant change and I would like more testing (despite the fact I have tried to unit-test this as much as possible).

New Core with svo_value is here:

svn://www.ultimatepp.org/upp/branches/svo_value

and I would like to ask everybody, for the sake of breaking/non-breaking your code, please try to checkout this, compile your mission critical (or not) projects with this new Value implementation and report any irregularities!

I think I will be finally merging svo_value sometimes in March. So please, test before that

Mirek
