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 exept 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 assignements (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

Subject: Re: Refactored Value coming....
Posted by cbpporter on Mon, 16 Jan 2012 11:13:01 GMT
View Forum Message <> Reply to Message

I'll check it out. Sound good.

But I must ask: how did you fit a 16 byte String into a 16 byte Value?

Subject: Re: Refactored Value coming....

Posted by mirek on Mon, 16 Jan 2012 14:34:10 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Mon, 16 January 2012 06:13I'll check it out. Sound good.

But I must ask: how did you fit a 16 byte String into a 16 byte Value?

By adding something to the String, obviously

Mirek

Subject: Re: Refactored Value coming....
Posted by kohait00 on Tue, 17 Jan 2012 10:45:16 GMT
View Forum Message <> Reply to Message

hi mirek..

sounds really good..
did you spend any time on easying the export of custom objects to Value too? remember Null handling

cheers

Subject: Re: Refactored Value coming....
Posted by mirek on Tue, 17 Jan 2012 10:54:31 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Tue, 17 January 2012 05:45hi mirek..

sounds really good..
did you spend any time on easying the export of custom objects to Value too? remember Null handling

cheers

Null handling stays at your proposed SetNull.

I would say that things are not worst that they were (and it is backward compatible too).

Mirek

Subject: Re: Refactored Value coming....
Posted by koldo on Tue, 17 Jan 2012 11:25:54 GMT
View Forum Message <> Reply to Message

Hello Mirek

I have got these errors just after including new Core files from svo_value:

In Sql/Sql.h I get this error:
c:\upp\uppsrc\sql\Sqls.h(27) : error C2660: 'Upp::String0::IsString' : function does not take 1 arguments

Perhaps IsString(q) would have to be changed to ::IsString(q).

In addition I get this in Core/Defs.h:
C:\upp\uppsrc\Core/Defs.h(353) : error C2679: binary '=' : no operator found which takes a right-hand operand of type 'const Upp::Nuller' (or t
 here is no acceptable conversion)
	c:\upp\uppsrc\sql\Sqls.h(31): could be 'Upp::SqlRaw &Upp::SqlRaw::operator =(const Upp::SqlRaw &)'
	while trying to match the argument list '(Upp::SqlRaw, const Upp::Nuller)'
	c:\upp\uppsrc\core\Value.hpp(203) : see reference to function template instantiation 'void Upp::SetNull<T>(T &)' being compiled
	with
	[
	   T=Upp::SqlRaw
	]
	c:\upp\uppsrc\core\Value.hpp(224) : see reference to function template instantiation 'const T &Upp::GetStaticNull<T>(void)' being compi
 led
	with
	[
	   T=Upp::SqlRaw
	]
	c:\upp\uppsrc\core\Value.hpp(304) : see reference to function template instantiation 'const T &Upp::Value::Get<T>(void) const' being co
 mpiled
	with
	[
	   T=Upp::SqlRaw
	]
	c:\upp\uppsrc\core\Value.hpp(304) : while compiling class template member function 'const Upp::SqlRaw &Upp::RawValue<T>::Extract(const
 Upp::Value &)'
	with
	[
	   T=Upp::SqlRaw
	]
	c:\upp\uppsrc\sql\Sqls.h(24) : see reference to class template instantiation 'Upp::RawValue<T>' being compiled
	with
	[

T=Upp::SqlRaw
]

Subject: Re: Refactored Value coming....
Posted by mirek on Tue, 17 Jan 2012 12:48:17 GMT
View Forum Message <> Reply to Message

koldo wrote on Tue, 17 January 2012 06:25Hello Mirek

I have got these errors just after including new Core files from svo_value:

In Sql/Sql.h I get this error:
c:\upp\uppsrc\sql\Sqls.h(27) : error C2660: 'Upp::String0::IsString' : function does not take 1 arguments

This one should already be fixed.

Quote:
In addition I get this in Core/Defs.h:
C:\upp\uppsrc\Core/Defs.h(353) : error C2679: binary '=' : no operator found which takes a right-hand operand of type 'const Upp::Nuller' (or t
here is no acceptable conversion)
c:\upp\uppsrc\sql\Sqls.h(31): could be 'Upp::SqlRaw &Upp::SqlRaw::operator =(const Upp::SqlRaw &)'
while trying to match the argument list '(Upp::SqlRaw, const Upp::Nuller)'
c:\upp\uppsrc\core\Value.hpp(203) : see reference to function template instantiation 'void Upp::SetNull<T>(T &)' being compiled
with
[
T=Upp::SqlRaw
]
c:\upp\uppsrc\core\Value.hpp(224) : see reference to function template instantiation 'const T &Upp::GetStaticNull<T>(void)' being compi
led
with
[
T=Upp::SqlRaw
]
c:\upp\uppsrc\core\Value.hpp(304) : see reference to function template instantiation 'const T &Upp::Value::Get<T>(void) const' being co
mpiled
with
[
T=Upp::SqlRaw
]
c:\upp\uppsrc\core\Value.hpp(304) : while compiling class template member function 'const

Upp::SqlRaw &Upp::RawValue<T>::Extract(const
 Upp::Value &)'
        with
        [
            T=Upp::SqlRaw
        ]
        c:\upp\uppsrc\sql\Sqls.h(24) : see reference to class template instantiation
'Upp::RawValue<T>' being compiled
        with
        [
            T=Upp::SqlRaw
        ]
[/quote]

Not sure what this means; so far I was using svo_value to compile ide (and I am running it now) in Win32 and for skylark development in Linux64 (there I have noticed and fixed the first bug). I have not encountered the second issue so far. Perhpas it depend on code compiled?

Mirek

Subject: Re: Refactored Value coming....
Posted by koldo on Tue, 17 Jan 2012 13:28:44 GMT
View Forum Message <> Reply to Message

Hello Mirek

Perhaps second problem comes from the solution I did for first problem.

I will update today the sources and I will try it again.

Subject: Re: Refactored Value coming....
Posted by koldo on Wed, 18 Jan 2012 08:17:41 GMT
View Forum Message <> Reply to Message

Hello Mirek

The first problem still remains for me.

Subject: Re: Refactored Value coming....
Posted by koldo on Thu, 19 Jan 2012 10:39:06 GMT
View Forum Message <> Reply to Message

Hello Mirek

Solved!. In a first test my main programs work perfectly.

Subject: Re: Refactored Value coming....
Posted by mirek on Mon, 30 Jan 2012 14:56:06 GMT
View Forum Message <> Reply to Message

After noticing that branch diverged to the point that svn is not able to merge trunk into it anymore, I have manually reintegrated svo_value branch into trunk as conditional compilation code. You can now add "SVO_VALUE" flag to main config and trunk provides the new Value implementation.