

Infrastructure - Task #7852

Story # 7832 (In Progress): migrate from JibX to JAXB for XML binding / codegen

compare performance vs JibX

2016-07-26 07:28 - Rob Nahf

Status:	Closed	Start date:	2016-07-26
Priority:	Normal	Due date:	
Assignee:	Rob Nahf	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		Story Points:	
Milestone:	None		
Product Version:	*		

Description

JAXB is reportedly slower, as it uses reflection to set up the marshalling context (for each type). the Marshaller and Unmarshaller are not thread safe, but are somewhat expensive to build and throw away. (GC)

TypeMarshaller, for now, has both JAXB and JiBX implemented, to enable comparisons (USE_JIBX property) also has CACHE_CONTEXT and CACHE_MARSHALLERS switches.

Associated revisions

Revision 18250 - 2016-07-26 23:24 - Rob Nahf

refs #7846: refactored types to use new DateMarshallingAdapter. refs #7852: performance testing. Added Marshaller, Unmarshaller caching logic to TypeMarshaller for performance testing, as well as performance tests to TypeMarshallerTestCase.

History

#1 - 2016-07-26 15:39 - Rob Nahf

Using a 7000 item ObjectList, created a unit test that profiles performance of unmarshalling and then remarshalling, repeated 50x.

Test results are the mean and median.

Both JiBX and JAXB have initialization costs associated with them, so I list first and second iteration results. The pairs are the unmarshalling / marshalling values for each line.

JiBX:

first iteration 376 / 184
second 215 / 80
median 42 / 17

JAXB (no caching)

first iteration 700 / 302
second 231 / 117
median 63 / 38

JAXB - caching Context

first iteration 628 / 266
second 203 / 85
median 41 / 19

JAXB - caching Context and (Un)Marshallers

first iteration 657 / 2279
second 177 / 69
median 42 / 20

There is a 50% improvement in caching the Context, but no appreciable difference if we cache themarshallers and unmarshallers themselves.

Caching the JAXB context makes the median unmarshalling and marshalling times equivalent to JiBX, although I'm surprised that not caching the context didn't cause a bigger performance issue. I would have expected no improvement in performance over each iteration. Perhaps the JAXB implementations have some optimizations in built in.

#2 - 2016-07-27 00:10 - Rob Nahf

Since caching themarshallers and unmarshallers don't add any performance gain, we should remove it to simplify the code base. As per a member of the JAXB 2 expert group (<http://stackoverflow.com/a/7400735>):

"JAXBContext is thread safe and should only be created once and reused to avoid the cost of initializing the metadata multiple times. Marshaller and Unmarshaller are not thread safe, but are lightweight to create and could be created per operation."

#3 - 2016-08-18 21:44 - Rob Nahf

- *Status changed from In Progress to Closed*

- *% Done changed from 30 to 100*

- *translation missing: en.field_remaining_hours set to 0.0*

Removed marshalling and unmarshalling caching, since it did not affect performance.