

Mulgara - Bug #137

xsd:decimal Datatype Not Preserved

08/08/2008 05:30 PM - James -

Status:	Closed	Start date:				
Priority:	Urgent	Due date:				
Assignee:	Paula Gearon	% Done:	0%			
Category:	Mulgara	Estimated time:	0.00 hour			
Target version:						
Resolution:	fixed					
Description						
Any literal with a datatype of xsd:decimal is converted into another datatype depending on the numeric value of its label. This can cause other applications that depend on the datatype to break. For example, JAXB maps xsd:decimal to java.math.BigDecimal.						
http://en.wikipedia.org/wiki/JAXB						

History

#1 - 08/12/2008 04:09 PM - Paula Gearon

We will need to introduce a new type ID, and not rely on resolving to a concrete type.

#2 - 08/26/2008 05:51 AM - Paula Gearon

- Status changed from New to Closed
- Resolution set to fixed

SPDecimalImpl has now been changed to an abstract class. xsd:decimal is managed with one implementation: SPDecimalBaseImpl, while all extensions are handled with another implementation: SPDecimalExtImpl.

SPDecimalBaseImpl is encoded as a string, and is managed internally as a BigDecimal. SPDecimalExtImpl is encoded as a long, and is managed internally as a Long. It may be desirable to change some of the types to BigInteger, but this is preferable if we can use it.

#3 - 08/26/2008 06:07 AM - ronald -

Fixed in [r1177](#).

#4 - 08/26/2008 01:26 PM - James -

- Status changed from Closed to Feedback
- Resolution deleted (fixed)

The fix row causes this error when comparing the two subclasses.

```
Caused by: java.lang.IndexOutOfBoundsException: 8
at java.nio.HeapByteBuffer.get(HeapByteBuffer.java:121)
at org.mulgara.store.stringpool.xa.SPDecimalExtImpl$SPDecimalExtComparator.compare(SPDecimalImpl.java:412)
at org.mulgara.store.stringpool.xa.XAStringPoolImpl$Phase$SPAVLComparator.compare(XAStringPoolImpl.java:2763)
at org.mulgara.store.xa.AVLNode.findDown(AVLNode.java:294)
at org.mulgara.store.xa.AVLFile$Phase.find(AVLFile.java:371)
at org.mulgara.store.stringpool.xa.XAStringPoolImpl$Phase.findGNode(XAStringPoolImpl.java:1706)
at org.mulgara.store.stringpool.xa.XAStringPoolImpl.findGNode(XAStringPoolImpl.java:415)
at org.mulgara.resolver.StringPoolSession.localizeSPObject(StringPoolSession.java:544)
at org.mulgara.resolver.StringPoolSession.localize(StringPoolSession.java:455)
at org.mulgara.resolver.StringPoolSession.localizePersistent(StringPoolSession.java:210)
at org.mulgara.resolver.store.StatementStoreResolver.localizePersistent(StatementStoreResolver.java:451)
at org.mulgara.resolver.spi.TripleSetWrapperStatements.localize(TripleSetWrapperStatements.java:179)
at org.mulgara.resolver.spi.TripleSetWrapperStatements.getObject(TripleSetWrapperStatements.java:173)
at org.mulgara.resolver.store.StatementStoreResolver.modifyModel(StatementStoreResolver.java:356)
at org.mulgara.resolver.InternalResolver.modifyModel(InternalResolver.java:164)
at org.mulgara.resolver.ModifyModelOperation.doModify(ModifyModelOperation.java:247)
at org.mulgara.resolver.ModifyModelOperation.execute(ModifyModelOperation.java:208)
at org.mulgara.resolver.MulgaraInternalTransaction.execute(MulgaraInternalTransaction.java:623)
... 36 more
```

#5 - 08/26/2008 01:43 PM - James -

Adding "3^^xsd:decimal /after/ "3^^xsd:int causes the two literals to be folded into one.

#6 - 08/26/2008 08:30 PM - Paula Gearon

- Status changed from Feedback to Closed
- Resolution set to fixed

(In [r1182](#)) Updated SPComparator to take the subtype ID when comparing, so that these may be distinguished when being inserted. Only the Decimal objects will use this, as no other types have subtypes, however all implementations of the comparator needed to be updated to deal with this. This should close [#137](#)