

## EDIT - bug #10101

### Correct mapping and handling of sorted tree structures

07/28/2022 09:25 AM - Andreas Müller

<b>Status:</b>	In Progress	<b>Start date:</b>	
<b>Priority:</b>	Priority14	<b>Due date:</b>	
<b>Assignee:</b>	Andreas Müller	<b>% Done:</b>	20%
<b>Category:</b>	cdmlib	<b>Estimated time:</b>	0:00 hour
<b>Target version:</b>	Release 5.45	<b>Found in Version:</b>	
<b>Severity:</b>	normal		
<b>Description</b>			
<p>We currently have 3 sorted tree structures in the CDM: Classifications, TermTrees and PolytomousKeys.</p> <p>The correct mapping and handling of sorted tree structures is difficult. Children are modelled as Lists which usually is best handled with an <code>@OrderColumn</code> annotation (<a href="#">#3722</a>) and a bidirectional mapping with <code>orphanRemoval</code>. If <code>orphanRemoval</code> is not used orphaned (deleted) children need to be deleted manually, otherwise they stay in the database and, in the worst case, have the same <code>parent_id</code> and <code>sortindex</code> if <code>parent_id</code> is not correctly removed by bidirectionality implementation. This leads to kind of duplicates and, even worse, to gaps in the <code>sortindex</code> when the tree structure is persisted via merge. Gaps in the <code>sortindexes</code> (e.g. 0 followed by 3) lead to null values when the list is reloaded (<a href="#">#5536</a>, <a href="#">#8127</a>).</p> <p>However, though <code>orphanRemoval</code> handles this best for Lists it is not good for (recursive) trees because moving a subtree to another parent then requires to clone the full subtree as not only the root but also all its children will be automatically deleted by hibernate via <code>orphanRemoval</code>. For large trees this is unwanted behavior and also for smaller trees cloning is not a great solution.</p> <p>So a better solution is to manually remove the nodes if they are not used anymore. We tried to do this automatically for merge via the <code>PostMergeEntityListener</code> but still without success. See comments in <a href="#">#note-7</a>.</p> <p>Therefore a new merge method was created to merge together with the entities that should be deleted. This seems to work.</p> <p>Note: To avoid the <code>sortindex</code> still has a gap for some reason we have to touch the children list somehow before commit. This is currently done in the <code>PostMergeEntityListener</code>.</p> <p>Related tests are: <code>PolytomousKeyNodeServiceTest.testMergeXXX()</code> (maybe will move to <code>PolytomousKeyServiceTest</code> partly)</p>			
<b>Related issues:</b>			
Related to EDIT - task <a href="#">#3722</a> : Using <code>@OrderColumn</code> does not work in trees		<b>Closed</b>	<b>05/23/2022</b> <b>05/27/2022</b>
Related to EDIT - task <a href="#">#8127</a> : [Reminder] to check if Hibernate Null Value bug...		<b>Discussed</b>	<b>05/23/2022</b> <b>05/27/2022</b>
Related to EDIT - bug <a href="#">#5211</a> : Error when creating new taxon /problem with sort...		<b>In Progress</b>	

#### Associated revisions

##### Revision [8b07f256](#) - 07/28/2022 02:05 PM - Andreas Müller

ref #10101 add some javadoc

##### Revision [8f79b3aa](#) - 07/28/2022 04:47 PM - Andreas Müller

ref #3722, ref #8127, ref #5536, ref #10101 adapt test for removing PKNode and add new test for moving node and add LIE handling for node removal

##### Revision [e6cd93a9](#) - 07/28/2022 05:25 PM - Andreas Müller

ref #10101 add commented implementation for trying to handle orphan removal in the `PostMergeEntityListener`

##### Revision [cf36c59d](#) - 07/28/2022 05:52 PM - Andreas Müller

ref #10101 fix possible null value in `refreshNodeNumbering`

##### Revision [93d9d454](#) - 07/28/2022 06:23 PM - Andreas Müller

ref #10101 fix `sortindex` gaps in polytomous key merge

##### Revision [2c08ec06](#) - 07/28/2022 06:24 PM - Andreas Müller

ref #10101 add `sortindex` fix to taxon node and term node, too

**Revision 96326c59 - 07/28/2022 07:27 PM - Andreas Müller**

ref #10101 adapt PolytomousKeyNodeServiceTest to new orphaned delete method in services

**Revision 9f707aab - 07/28/2022 08:02 PM - Andreas Müller**

ref #10101 preliminary remove children initialization in PostMergeEntityListener

**Revision 031c8642 - 07/28/2022 08:36 PM - Andreas Müller**

ref #10101 add the childNodes touche for TaxonNode and TermNode again in PostMergeEntityListener

**Revision ccd3aaab - 07/28/2022 09:34 PM - Andreas Müller**

ref #10101 again preliminary remove children initialization in PostMergeEntityListener

**Revision 4efb0c4a - 07/28/2022 10:13 PM - Andreas Müller**

ref #10101 catch and log no-session exception in PostMergeEntityListener instead of removing initialization completely

**Revision d5e92c4b - 07/29/2022 01:23 PM - Andreas Müller**

ref #10101 add (partly ignored) tests to TaxonNodeServiceImplTest and TermNodeServiceImplTest

**Revision 63e1c600 - 07/29/2022 01:25 PM - Andreas Müller**

ref #10101 add saveDetached tests to PolytomousKeyNodeServiceTest (and replace save by merge in other tests)

**Revision 520f2d6d - 07/29/2022 01:55 PM - Andreas Müller**

ref #10101 add testSaveDetachedWithMove test to PolytomousKeyNodeServiceTest

**Revision 3ad25ffd - 07/29/2022 01:57 PM - Andreas Müller**

ref #10101 removed cascade from TermRelationBase.graph, not needed and probably also not correct

**Revision e5477781 - 07/29/2022 02:00 PM - Andreas Müller**

ref #10101 removed cascade from TermNode.parent, not needed and probably also not correct

**Revision cb923587 - 07/29/2022 02:04 PM - Andreas Müller**

ref #10101, ref #8127, ref #3722, ref #5536 remove HHH\_9751\_Util class as it is not used anymore

**Revision 1aa6c88f - 07/29/2022 02:13 PM - Andreas Müller**

ref #10101 delegat onMerge(event) to onMerge(event, alreadyCopied) in PostMergeEntityListener

**Revision 09abc656 - 07/29/2022 03:31 PM - Andreas Müller**

ref #10101 remove HHH\_9751 from TaxEditor

## History

---

**#1 - 07/28/2022 09:27 AM - Andreas Müller**

- Related to task #3722: Using @OrderColumn does not work in trees added

**#3 - 07/28/2022 09:27 AM - Andreas Müller**

- Related to task #8127: [Reminder] to check if Hibernate Null Value bug still exists added

**#4 - 07/28/2022 09:31 AM - Andreas Müller**

- Related to bug #5211: Error when creating new taxon /problem with sortindex in taxon tree added

**#6 - 07/28/2022 05:31 PM - Andreas Müller**

- Description updated

**#7 - 07/28/2022 05:33 PM - Andreas Müller**

Code comment from PostMergeEntityListener.java:

```
// #10101 the following code tried to handle orphanRemoval for key nodes that were  
// really removed from the graph. Generally the removal worked but it was not possible at this
```

```
// place to guarantee that the node which was removed from the parent was not used elsewhere
// (has a new parent). For this one needs to retrieve the new state of the node (or of its new parent).
// But this is not difficult or impossible at this point as the node is not part of the graph to
// to be merged or if it is because its new parent is part of the graph also, it is not guaranteed
// that it has already treated so far.
// Maybe it can better be handled by another type of listener therefore I keep this code here.
// See #10101 for further information on this issue and how it was solved.
// The implementation was partly copied from https://stackoverflow.com/questions/812364/how-to-determine-colle
ction-changes-in-a-hibernate-postupdateeventlistener
```

<https://dev.e-taxonomy.eu/gitweb/cdmlib.git/blob/e1284d9aace334813f3bee700c1aa1127033f1ef/cdmlib-persistence/src/main/java/eu/etaxonomy/cdmlib/persistence/hibernate/PostMergeEntityListener.java>

#### #8 - 07/28/2022 08:54 PM - Andreas Müller

- Description updated

#### #9 - 07/29/2022 01:36 PM - Andreas Müller

- Status changed from New to In Progress

- Priority changed from New to Priority14

- Target version changed from Release 5.32 to Release 5.33

- % Done changed from 0 to 20

I added tests for mergeDetached and saveDetached on nodes and trees and with removal and move of nodes to PolytomousKeyNodeServiceTest, TermNodeServiceTest and TaxonNodeServiceTest.

- for PolytomousKeys all tests run well and also the TaxEditor does not seem to have problems anymore.
- for TermNodes the mergeDetached on node with removal does not work for some reason, though the configuration is very similar to the one for PolytomousKeys, however, in TaxEditor it seems to work, maybe because we use DTOs there and finde graine operations (either delete or create)
- for TaxonNodes the tests still need to be adapted as TaxonNodes, on purpose, do not cascade. However, TaxEditor seems to work with (probably same reason as for TermNodes)

All failing tests set to @Ignore, so there are open issues. Especially the TermNodes issue needs to be further investigated.

Also I keep #8127 open as the current implementation works only with the workaround in PostMergeEntityListener and does not work at all yet for TermNodes and is not tested for TaxonNodes

#### #10 - 07/29/2022 02:05 PM - Andreas Müller

I removed the HHH\_9751\_Util class as it is currently not used anymore, please recreate from [cb923587ce746](#) and taxeditor commit 09abc6569 if it is used in future again.

#### #11 - 07/29/2022 02:15 PM - Andreas Müller

Comment from onMerge in PostMergeEntityListener:

```
Added in context of working on #10101, it was unclear to me
why this method was not implemented before. Debugging the merge process
revealed that this method was primarily called on the first entity
to merge (as copied already is not necessary then).
So setting the ID and adding to the newEntitiesMap seems
not to be implemented for entities handle only here.
Needs intensive testing if there is not maybe a reason why the method
was not implemented.
```

#### #12 - 07/29/2022 05:47 PM - Andreas Müller

Finding duplicates:

```
SELECT pkn.key_id, pkn.parent_id, pkn.sortindex, COUNT(*) AS n
FROM PolytomousKeyNode pkn
GROUP BY pkn.key_id, pkn.parent_id, pkn.sortindex
HAVING n > 1;
```

```
SELECT tn.classification_id, tn.parent_id, tn.sortindex, COUNT(*) AS n
FROM TaxonNode tn
GROUP BY tn.classification_id, tn.parent_id, tn.sortindex
HAVING n > 1;
```

```
SELECT tr.graph_id, tr.parent_id, tr.sortindex, COUNT(*) AS n
FROM TermRelation tr
```

```
GROUP BY tr.graph_id, tr.parent_id, tr.sortindex  
HAVING n > 1;
```

A helpful SQL for checking the correctness of the sortindex (here for PKNode):

```
SELECT pkn.id, pkn.created, pkn.updated, nodenumber, key_id, parent_id, sortindex  
FROM PolytomousKeyNode pkn  
WHERE key_id = 24  
ORDER BY key_id, parent_id, id, sortindex;
```

```
SELECT pkn.id, pkn.created, pkn.updated, nodenumber, KEY_id, parent_id, sortindex, pkn.revtype  
FROM PolytomousKeyNode_AUD pkn  
WHERE key_id = 24 OR id = 394  
ORDER BY key_id, parent_id, id, sortindex
```

**#13 - 09/08/2022 11:47 AM - Andreas Müller**

- *Tags set to 5.32-Remaining*

- *Target version changed from Release 5.33 to Release 5.45*