EDIT - bug #8874

move taxon is reverted when taxon is edited afterwards

03/05/2020 09:40 AM - Katja Luther

Status:	Resolved	**Start date:**	
Priority:	Highest	**Due date:**	
Assignee:	Andreas Müller	**% Done:**	70%
Category:	taxeditor	**Estimated time:**	0:00 hour
Target version:	Release 5.19		
Severity:	normal	**Found in Version:**	

Description

mail W.B.:

ein merkwürdiger Bug – ist mir schon mehrfach aufgefallen, aber ich habe ihn nicht so recht dokumentiert. Caryophyllales_spp:
Ich verschiebe ein als doubtful gekennzeichnetes Taxon (Alternanthera eupatorioides Remy) im Baum (über den Kontextdialog) – in diesem Fall aus der Gattung Alternanthera unter das Pseudotaxon _Amaranthaceae excluded. Dort öffne ich's im Node-Dialog, markiere es als excluded und gebe meinen Kommentar ein.
Wenn ich es jetzt wieder im Name Editor aufmache und dort etwas ändere (z.B. [früher aufgefallen] die doubtful flag entferne oder [jetzt] die sec.-Ref. ändere, dann wird das excluded flag entfernt und das Taxon springt zurück an seinen ursprünglichen Ort. Der Kommentar zu excluded ist noch da, aber das flag ist weg.
Taxon wieder verschoben, Flag wieder gesetzt, Name Editor, Änderung (diesmal das doubtful-flag entfernt) – speichern – Alles gut.

Related issues:

Related to EDIT - feature request #9420: Harmonize Update- and ImportResult	**New**

Associated revisions

Revision 136d38e9 - 09/28/2020 12:53 PM - Katja Luther

ref #8874: add taxonnode to getUpdatedTaxon

History

#1 - 03/09/2020 04:26 PM - Andreas Müller

- Target version changed from Release 5.13 to Release 5.14

#2 - 04/08/2020 03:46 PM - Andreas Müller

- Target version changed from Release 5.14 to Release 5.15

#3 - 06/23/2020 01:51 PM - Andreas Müller

- Target version changed from Release 5.15 to Release 5.18

#4 - 09/28/2020 12:53 PM - Katja Luther

- Assignee changed from Katja Luther to Andreas Müller

this should be fixed. please review.

#5 - 12/03/2020 06:09 PM - Andreas Müller

- Assignee changed from Andreas Müller to Katja Luther

- Target version changed from Release 5.18 to Release 5.19

Katja, do you remember why this ticket was assigned to me? Simply a mistake?
Can we reproduce this behavior?

#6 - 01/15/2021 10:25 AM - Katja Luther

- Status changed from New to Resolved

- Assignee changed from Katja Luther to Andreas Müller

this ticket was solved and assigned to you for review, sorry forgot to set to resolved.

#7 - 01/21/2021 03:35 PM - Andreas Müller

I think I remember the mail discussion about this issue. There, we discussed that this happens only if the taxon was opened before. Is this correct?
As it differs from the above description "Wenn ich es jetzt wieder im Name Editor aufmache" we should store such information in the ticket, too. This makes it easier to review (and implement) and avoids false "WorksForMe"s.

#8 - 01/21/2021 03:55 PM - Andreas Müller

It seems to work. However, I do not fully understand, why. Don't the changes try to guarantee that all open taxa related to the move are saved before?
So, obviously they are not saved. Is this what is expected? (I mean it is good if they do not need to be saved, I want to understand how it works)

#9 - 01/21/2021 03:58 PM - Andreas Müller

I also wonder, why in the code only the UpdateResult part was adapted, but not the ImportResult. Is this specific to UpdateResult?

If not, we should try to remove code redundancy by moving the code

```
                Taxon taxon = null;

                if (object instanceof Taxon){
                    taxon = HibernateProxyHelper.deproxy(object, Taxon.class);
                }else if (object instanceof Synonym){
                    Synonym syn = HibernateProxyHelper.deproxy(object, Synonym.class);
                    taxon = syn.getAcceptedTaxon();
                }else if (object instanceof TaxonNode){
                    taxon = ((TaxonNode)object).getTaxon() != null? ((TaxonNode)object).getTaxon():null;
                }
                if (taxon != null){
                    taxaToUpdate.add(taxon);
                }
```

to a separate method.

#10 - 01/21/2021 04:00 PM - Andreas Müller

- Status changed from Resolved to Feedback

- Assignee changed from Andreas Müller to Katja Luther

- % Done changed from 0 to 70

Another more critical issue related to the code reviewed:

The following code:

```
for (Map.Entry<String, Integer> object: result.entrySet()){
                Taxon taxon = null;
                if (object instanceof Taxon){
```

to me looks strange. How can "object" be of type Taxon if it is a map.
I don't think this can work. Has it ever been tested?

#11 - 01/25/2021 09:39 AM - Katja Luther

Andreas Müller wrote:

> Another more critical issue related to the code reviewed:
>
> The following code:
>
> ```
> for (Map.Entry<String, Integer> object: result.entrySet()){
>                 Taxon taxon = null;
>                 if (object instanceof Taxon){
> ```
>
> to me looks strange. How can "object" be of type Taxon if it is a map.
> I don't think this can work. Has it ever been tested?

I had a look and this part is nonesense... the updatedRecords only count the updated records.

I would remove the code.

#12 - 01/25/2021 09:43 AM - Katja Luther

Andreas Müller wrote:

> I also wonder, why in the code only the UpdateResult part was adapted, but not the ImportResult. Is this specific to UpdateResult?
>
> If not, we should try to remove code redundancy by moving the code
>
> ```
>                 Taxon taxon = null;
>
>                 if (object instanceof Taxon){
>                     taxon = HibernateProxyHelper.deproxy(object, Taxon.class);
>                 }else if (object instanceof Synonym){
>                     Synonym syn = HibernateProxyHelper.deproxy(object, Synonym.class);
>                     taxon = syn.getAcceptedTaxon();
>                 }else if (object instanceof TaxonNode){
>                     taxon = ((TaxonNode)object).getTaxon() != null? ((TaxonNode)object).getTaxon():null;
>                 }
>                 if (taxon != null){
>                     taxaToUpdate.add(taxon);
>                 }
> ```
>
> to a separate method.

The import result does not contains specific information about the updated or new elements therefore I would remove the code as mentioned above.

Maybe we should think about a more detailed ImportResult, but therefore we need to implement it in cdmlib/io.

#13 - 01/25/2021 10:22 AM - Andreas Müller

Katja Luther wrote:

> Andreas Müller wrote:
>
> > I also wonder, why in the code only the UpdateResult part was adapted, but not the ImportResult. Is this specific to UpdateResult?
> >
> > If not, we should try to remove code redundancy by moving the code
> >
> > ```
> >                 Taxon taxon = null;
> >
> >   if (object instanceof Taxon){
> >   taxon = HibernateProxyHelper.deproxy(object, Taxon.class);
> >   }else if (object instanceof Synonym){
> >   Synonym syn = HibernateProxyHelper.deproxy(object, Synonym.class);
> >   taxon = syn.getAcceptedTaxon();
> >   }else if (object instanceof TaxonNode){
> >   taxon = ((TaxonNode)object).getTaxon() != null? ((TaxonNode)object).getTaxon():null
> > ;
> >   }
> >   if (taxon != null){
> >   taxaToUpdate.add(taxon);
> >   }
> > ```
> >
> > to a separate method.
>
> The import result does not contains specific information about the updated or new elements therefore I would remove the code as mentioned above.

By "moving the code" I did not mean "remove" but extract the code to a separate method and call it 2x (so the typical thing to avoid redundancy). Did you check if it is really not used for imports at all? Otherwise we need the ImportResult part and can't simly remove it, or?

#14 - 01/25/2021 10:28 AM - Katja Luther

Andreas Müller wrote:

> Katja Luther wrote:
>
> > Andreas Müller wrote:
> >
> > > I also wonder, why in the code only the UpdateResult part was adapted, but not the ImportResult. Is this specific to UpdateResult?
> > >
> > > If not, we should try to remove code redundancy by moving the code
> > >
> > > ```
> > >                 Taxon taxon = null;
> > >
> > >   if (object instanceof Taxon){
> > > ```

> > > ```
> > >   taxon = HibernateProxyHelper.deproxy(object, Taxon.class);
> > >   }else if (object instanceof Synonym){
> > >   Synonym syn = HibernateProxyHelper.deproxy(object, Synonym.class);
> > >   taxon = syn.getAcceptedTaxon();
> > >   }else if (object instanceof TaxonNode){
> > >   taxon = ((TaxonNode)object).getTaxon() != null? ((TaxonNode)object).getTaxon():
> > > null;
> > >   }
> > >   if (taxon != null){
> > >   taxaToUpdate.add(taxon);
> > >   }
> > > ```
> > >
> > > to a separate method.
> >
> > The import result does not contains specific information about the updated or new elements therefore I would remove the code as mentioned above.
>
> By "moving the code" I did not mean "remove" but extract the code to a separate method and call it 2x (so the typical thing to avoid redundancy).
> Did you check if it is really not used for imports at all? Otherwise we need the ImportResult part and can't simly remove it, or?

I know that you meant move not remove but this method only get the information about which taxa where updated and updates the editors.

The import result only contains informations about the type of elements and how many of them are updated. This information is used for the report but can not be used for updating. As I already mentioned, should we think about a more detailed ImportResult.

#15 - 01/25/2021 10:29 AM - Andreas Müller

Katja Luther wrote:

> Maybe we should think about a more detailed ImportResult, but therefore we need to implement it in cdmlib/io.

Yes, we definitely should unify ImportResult and UpdateResult somehow, e.g. by having a common base class. I think we have talked about this alrady. Can you have a look, if there is a ticket already and crate one if not?
May we discuss this in standup later on?

#16 - 01/25/2021 03:47 PM - Katja Luther

Andreas Müller wrote:

> It seems to work. However, I do not fully understand, why. Don't the changes try to guarantee that all open taxa related to the move are saved before?
> So, obviously they are not saved. Is this what is expected? (I mean it is good if they do not need to be saved, I want to understand how it works)

Yes, but after the operation the updated taxa need to be loaded again in the already open editors, otherwise the taxon would overwrite the changes of the serverside operation.

#17 - 01/25/2021 03:59 PM - Katja Luther

- Status changed from Feedback to Closed

#18 - 01/25/2021 04:01 PM - Katja Luther

Andreas Müller wrote:

> Katja Luther wrote:
>
> > Maybe we should think about a more detailed ImportResult, but therefore we need to implement it in cdmlib/io.
>
> Yes, we definitely should unify ImportResult and UpdateResult somehow, e.g. by having a common base class. I think we have talked about this alrady. Can you have a look, if there is a ticket already and crate one if not?
> May we discuss this in standup later on?

There is no ticket handling UpdateResult and ImportResult, actually they do not have the same base class like Update- and DeleteResult. I create a new ticket about this issue. (#9420)

#19 - 01/25/2021 04:03 PM - Andreas Müller

- Status changed from Closed to Resolved

- Assignee changed from Katja Luther to Andreas Müller

I want to check again what I meant therefore I put it back to review to remind myself to do so.

#20 - 01/25/2021 04:04 PM - Katja Luther

- Related to feature request #9420: Harmonize Update- and ImportResult added