

Database - Bug #6703

loading schema table DMO interface with no defined fields

08/25/2022 05:33 AM - Constantin Asofiei

Status:	WIP	Start date:	
Priority:	Normal	Due date:	
Assignee:	Ovidiu Maxiniuc	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No		
vendor_id:	GCD		
Description			

History

#2 - 08/25/2022 05:34 AM - Constantin Asofiei

If the .df has a table with no fields defined, import or loading this DMO interface fails with this:

```
[java] Caused by: java.lang.ArrayIndexOutOfBoundsException: 0
[java]   at com.goldencode.p2j.persist.orm.RecordMeta.readPropertyMeta(RecordMeta.java:505)
[java]   at com.goldencode.p2j.persist.orm.RecordMeta.<init>(RecordMeta.java:227)
[java]   at com.goldencode.p2j.persist.orm.DmoClass.assembleImplementation(DmoClass.java:442)
[java]   at com.goldencode.p2j.persist.orm.DmoMetadataManager.registerDmo(DmoMetadataManager.java:241)
```

#3 - 08/26/2022 04:08 AM - Constantin Asofiei

A fix for the DMO interface loading (for runtime and DB import) is in 6129a/14273.

This does not solve all issues, CRUD operations still need to be checked, and import shows this (the .d for this 'no-field' table has no records, obviously):

```
[java] <table>.d: Error processing import data; 200 of 0 record(s) successfully processed; 0 record(s) uncommitted due to this error; -200 record(s) dropped.
[java] java.lang.ArrayIndexOutOfBoundsException: 0
[java]   at com.goldencode.p2j.persist.orm.Persister.insert(Persister.java:781)
[java]   at com.goldencode.p2j.persist.orm.Persister.bulkInsert(Persister.java:344)
[java]   at com.goldencode.p2j.persist.orm.Session.bulkSave(Session.java:739)
[java]   at com.goldencode.p2j.schema.ImportWorker$Library.importTable(ImportWorker.java:1390)
[java]   at com.goldencode.p2j.schema.ImportWorker$Library.lambda$importAsync$3(ImportWorker.java:1887)
[java]   at java.lang.Thread.run(Thread.java:750)
```

#4 - 09/08/2022 07:24 PM - Ovidiu Maxiniuc

- Assignee set to Ovidiu Maxiniuc
- Status changed from New to WIP

Constantin, I reviewed it and your fix in 6129a/14273 seems logical.

I also added two changes in 6129a/14387:

- Persister.java: to avoid `ArrayIndexOutOfBoundsException` when persisting records with no defined fields. This is the issue you discovered with the import, but also when a record is created in the application;
- `RecordBuffer.java`: allow unchanged transient records to be flushed even if no trigger is fired. Without this update, the transient records were never saved. I do not know how this case was overlooked.

Eric, please review this update, especially the second issue.

#5 - 09/19/2022 03:09 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

Constantin, I reviewed it and your fix in 6129a/14273 seems logical.

I also added two changes in 6129a/14387:

- `Persister.java`: to avoid `ArrayIndexOutOfBoundsException` when persisting records with no defined fields. This is the issue you discovered with the import, but also when a record is created in the application;

I don't understand this part. It looks like we are trying to insert an "empty" record in the case of a no-column table. What does that mean? How can a no-column table have any rows, since there are no values to insert? What is an "empty" record in this context and why would we want one in such a table?

- `RecordBuffer.java`: allow unchanged transient records to be flushed even if no trigger is fired. Without this update, the transient records were never saved. I do not know how this case was overlooked.

I think the changes here are ok.

Eric, please review this update, especially the second issue.

Consider my questions/comments above my review. Thanks.

#6 - 10/05/2022 02:03 PM - Ovidiu Maxiniuc

- % Done changed from 0 to 100

Eric Faulhaber wrote:

Ovidiu Maxiniuc wrote:

- Persister.java: to avoid `ArrayIndexOutOfBoundsException` when persisting records with no defined fields. This is the issue you discovered with the import, but also when a record is created in the application;

I don't understand this part. It looks like we are trying to insert an "empty" record in the case of a no-column table. What does that mean? How can a no-column table have any rows, since there are no values to insert? What is an "empty" record in this context and why would we want one in such a table?

Eric,

The changes were added as result of my experience with such tables on OE. These no-fields records can be created, and OE is aware of their existence. Even if the dump file is empty if no PSC footer is present, if it is generated, we can see the number of records in specific value.

Indeed, it may not be important at this moment, but later some queries might depend on the existence of these records. For example, can-find will return true if the records are present. If we do not create them, the execution path might be different then expected.

#7 - 10/05/2022 02:17 PM - Constantin Asofiei

Ovidiu, I tested loading a .d for a table with no fields in the Data Administration, in OpenEdge:

- created a no-field table
- inserted 20 records in it
- dumped the .d
- deleted all records
- tried to load the .d - got this error:

```
>> ERROR! Trailer indicated 20 records, but 0 records were loaded.
```

So there is no way to load records in a no-field table.

#8 - 10/05/2022 02:21 PM - Ovidiu Maxiniuc

My point was that the records existed in the original table and we need to have a table in FWD/SQL with same (well, equivalent) content. I was not referring to the process of import itself, but the final state.