

## Database - Bug #2337

### trigger scanning defect

07/10/2014 12:11 PM - Ovidiu Maxiniuc

<b>Status:</b> Closed	<b>Start date:</b>
<b>Priority:</b> High	<b>Due date:</b>
<b>Assignee:</b> Ovidiu Maxiniuc	<b>% Done:</b> 100%
<b>Category:</b>	<b>Estimated time:</b> 0.00 hour
<b>Target version:</b> Cleanup and Stablization for Server Features	<b>case_num:</b>
<b>billable:</b> No	
<b>vendor_id:</b> GCD	
<b>Description</b>	
<b>Related issues:</b>	
Related to Database - Feature #2339: generate schema-triggers.xml from Java a... <b>New</b>	

### History

#### #1 - 07/10/2014 12:16 PM - Ovidiu Maxiniuc

Because File.listFiles() does not work on jar locations, the initialization of the DatabaseTriggerManager (gathering the trigger classes for mapping to database objects) is broken and have to be rewritten.

A solution is using the intermediary xml file. This would also improved the startup time by locating the files directly. The difficulty is that the values from 'schema-triggers.xml' are progress-oriented because of the early stage when the file is created. Further rule-set pass on schema-triggers.xml later in schema conversion to create a derivative file needed at server startup.

Note:

This defect is not related [#2222](#) nor to Windows insensitive-case file lookup (see om\_upd20140707b.zip from #2332).

#### #2 - 07/10/2014 04:29 PM - Ovidiu Maxiniuc

- Status changed from New to WIP

Fixed the issue. It proved to be rather simple, just to add the class name (including package to current xml list of triggers) and save back (for the moment with another name) and load the xml when initiating the DatabaseTriggerManager. The new xml will be added to one of the application's jars (probably in the root of <customer>\_dmo ?).

However, when testing, I found that ASSIGN triggers are not converted correctly (they are treated as UI triggers).

LE: GES removed the customer name from this entry.

#### #3 - 07/10/2014 06:07 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

Fixed the issue. It proved to be rather simple, just to add the class name (including package to current xml list of triggers) and save back (for the moment with another name) and load the xml when initiating the DatabaseTriggerManager. The new xml will be added to one of the application's

jars (probably in the root of <customer>\_dmo ?).

I would not add this to the root of the jar, if it is associated with a particular schema (<db\_name>). What if a project has multiple schemas? Are all triggers combined from all schemas into a single XML file, or does each schema generate a separate XML file. If the latter, I would suggest putting the file in com/something/server/dmo/<db\_name>. If the former (all triggers across all schemas combined in one XML file), I would suggest that directory's parent.

However, when testing, I found that ASSIGN triggers are not converted correctly (they are treated as UI triggers).

How much effort to fix this?

LE: GES removed the customer-specific name, package path and database name from this entry.

#### #4 - 07/11/2014 05:00 AM - Ovidiu Maxiniuc

Eric Faulhaber wrote:

I would not add this to the root of the jar, if it is associated with a particular schema (<db\_name>). What if a project has multiple schemas? Are all triggers combined from all schemas into a single XML file, or does each schema generate a separate XML file. If the latter, I would suggest putting the file in com/something/server/dmo/<db\_name>. If the former (all triggers across all schemas combined in one XML file), I would suggest that directory's parent.

At the start of the middle part of conversion, all permanent databases are processed by schema/p2o.xml in a single tree, and temp-tables in the following step. That's the initial point when the schema-triggers.xml is created, dropping any old version of the file. Because of that, each trigger entry has an attribute with the database.

The utility of schema-triggers.xml is at the middle of back end within convert/base\_structure.xml. At this time, the file is processed from the trigger point of view, extracting the values for annotations and also updating the xml with the now-known class name of the trigger. I guess I'll use the second option. I will add the rule for copying the file to run/server and load it accordingly.

However, when testing, I found that ASSIGN triggers are not converted correctly (they are treated as UI triggers).

How much effort to fix this?

Probably it's a side-effect of something that has changed. Since the customer's server does not use session triggers (at least ASSIGNS), this did not surfaced until I run some basic tests for checking my update. I believe a couple of hours should be enough to find and fix this issue.

LE: GES removed the customer-specific name, package path and database name from this entry.

#### #5 - 07/11/2014 06:36 AM - Ovidiu Maxiniuc

I found the cause why ASSIGNS are not converted correctly.  
In the progress.g, in the on\_stmt function, the following was added:

```
// dbevent names can be used with widgets instead of buffers (customer code shows
// that at least DELETE can be used this way); so we must look ahead and only match
// here if this is really a database case
{ sym.isTable(LT(3).getText()) }?
```

The ASSIGN triggers are some exception in fact that they are related to a field, not a table/buffer. For my test-case for assign of book.isbn the LT is <DB\_SYMBOL> book.isbn, not a <SYMBOL> book as the antlr expects.  
I believe that a possible solution would be changing the condition to:

```
{ sym.isTable(LT(3).getText()) || sym.isField(LT(3).getText()) }?
```

#### #6 - 07/11/2014 11:22 AM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

I will add the rule for copying the file to run/server and load it accordingly.

I followed everything up to this point. Do you mean to put the loose file in the run/server directory instead of loading it from the <app\_name>\_dmo.jar file at runtime? We don't want to do this -- it should be loaded from the jar file.

LE: GES removed the customer app name from this entry.

#### #7 - 07/11/2014 11:30 AM - Ovidiu Maxiniuc

I see, for the customer, com/something/server/dmo, it make more sense.  
I was fooled by "directory's parent". I interpreted it as runtime directory, the xml file.

LE: GES removed the customer-specific package prefix for this entry.

#### #8 - 07/11/2014 11:32 AM - Eric Faulhaber

Ah, OK. Sorry for the confusion I caused.

**#9 - 07/11/2014 02:28 PM - Ovidiu Maxiniuc**

- File *om\_upd20140711a.zip* added
- Status changed from *WIP* to *Review*

Uploaded fix for trigger issues:

- the *schema-trigger.xml* improved in the second step
- *DatabaseTriggerManager* will open the xml from jar, parse and load triggers
- *progress.g* prepares *ASSIGN* triggers correctly for conversion
- *Utils* correctly recognize ignore-case paths on Windows

The update is merged with latest bzd revision (10567).

I am putting it to test. I believe that a conversion test and server startup should be enough for validating this update. (I am not expecting any changes in *majic* code, and only server-startup is affected).

**#10 - 07/11/2014 03:35 PM - Eric Faulhaber**

- Status changed from *Review* to *WIP*

Code review 20140711a:

The only comment I have is about the change in approach when initializing the trigger manager. I guess I didn't fully understand the nature of the change until I reviewed the code. I agree that reading the *schema-triggers.xml* file at startup should be much faster than scanning through all business logic class files for the *SchemaTrigger* annotation. My only concern is that this will make maintenance of an application just a little more difficult post-conversion, since developers will now have one more centralized artifact they need to remember to update when implementing a new trigger. Generally, we have been moving in the other direction with the introduction of Java annotations.

There are ways to mitigate the impact to startup time if it is an issue (e.g., the trigger manager could launch a temporary thread at server startup to do the scanning and block on trigger events until that thread completes).

If the primary driver for the change in approach was the failure of *File.listFiles* with a jar file, we can get around that as well. I can provide you with example code that reads files out of a directory hierarchy in a jar file.

That being said, the rest of the update looks good.

Greg, two things: what is your opinion on the above, and would you please review the parser change?

**#11 - 07/12/2014 01:43 PM - Eric Faulhaber**

I converted the customer's server project. The *schema-triggers.xml* file is left at the project root directory, but it is never copied to *src/com/something/server/dmo*, so it never ends up in *<app\_name>\_dmo.jar*.

If we ultimately decide to stay with this approach, we should do this as part of conversion (in *base\_structure.xml*), rather than delegating it to each application's *build.xml*. The customer's server app *build.xml* should already copy it to the *build/classes/...* directory, without the need for additional changes.

LE: GES replaced customer-specific application name and package path from this entry.

**#12 - 07/14/2014 07:15 AM - Ovidiu Maxiniuc**

- Status changed from WIP to Review

I see tree solutions now, not sure which is the best:

1. the easiest would be to keep the current one, implemented in 20140711a, with the addition of a rule for copying the schema-triggers.xml to classes during the conversion, instead of delegating this step to application's build.xml. (Eric's note-11)
2. 'manually' scanning the jar(s) for annotated classes. There are some aspects here:
  - iterating ZipEntry s from ZipInputStream, this might take some resources, but the solution is to do this in a temporary thread (note-10)
  - we don't know the jar that contains the classes, we could use some heuristically detection, or the application may choose to use multiple jars (for example, storing each database dmos/triggers in separate jars).
3. I have done some research for annotated class. I found a nice library Reflections that promises to handle the lookup of classes and annotations: <https://code.google.com/p/reflections/>. It might be useful in some other parts of our application but I am not sure if we want additional libraries to the project.

**#13 - 07/14/2014 10:19 AM - Ovidiu Maxiniuc**

- Status changed from Review to WIP

I noticed that unintentionally I changed the status of this issue back to "Review". Reverted to WIP.

**#14 - 07/14/2014 10:34 AM - Eric Faulhaber**

Let's keep all these alternatives in mind, but considering that it works well as is (I tested with the schema-triggers.xml file manually copied to the proper location), I want to leave it as is now. We can come back around and revert to a scanning approach if someone has a strong preference, but it is not high priority now.

Please fix conversion to copy the file to src/{application path}/dmo directory and post an update.

**#15 - 07/14/2014 10:37 AM - Eric Faulhaber**

Greg: when Ovidiu posts the next update, would you please review the parser change?

**#16 - 07/14/2014 10:50 AM - Greg Shah**

Code Review 0711a

I only reviewed the parser change. It looks good. Nice catch.

**#17 - 07/14/2014 10:56 AM - Greg Shah**

In regard to the annotations vs xml file approach question, my preference is for the annotations approach. If the performance of the server startup is an issue, then we can always write a tool to generate the xml "cache" file from the annotations, such that it could be part of the application build process and thus that file would always be up to date when changes are made to the application jar.

I'm fine with cleaning this up later, but you'd better add a task otherwise we will forget.

**#18 - 07/14/2014 02:16 PM - Eric Faulhaber**

- Assignee set to Ovidiu Maxiniuc
- % Done changed from 0 to 80
- Target version set to Milestone 11

Greg Shah wrote:

...we can always write a tool to generate the xml "cache" file from the annotations, such that it could be part of the application build process...

I like this idea a lot -- best of both worlds. I've opened [#2339](#) to track it as a potential, future improvement, no milestone.

Ovidiu: please continue with the current plan.

**#19 - 07/14/2014 02:17 PM - Eric Faulhaber**

- Project changed from Liberty to Database

**#20 - 07/14/2014 03:46 PM - Ovidiu Maxiniuc**

- File `om_upd20140714a.zip` added

Attached the update that saves the `schema-trigger.xml` to `src/{application path}/dmo` directory to be further packaged in jar. The file keeps current format, as Eric mentioned, the [#2339](#) will handle cleaning it up or writing needed infos to a new cache file.

I am putting this update also to test.

**#21 - 07/14/2014 10:32 PM - Eric Faulhaber**

Code review 0714a:

Looks good, please commit and distribute when this passes testing.

**#22 - 07/15/2014 02:24 PM - Ovidiu Maxiniuc**

There were no conversion differences (Except for the `schema-triggers.xml` file itself in `src/aero/timco/majic/dmo`).

There are a few failed tests:

6 in `ctrl+c / gso_ctrlc_3way_tests` (some killed client did the mess)

1 in `main / gso_tests`

7 in `main / tc_tests` (most of them cascading from clock jobs)

Re-run the tests.

**#23 - 07/25/2014 08:14 AM - Ovidiu Maxiniuc**

After lots of failing tests I asked for Constantin's opinion and he confirmed that the update is safe for commit. Luckily, the update was not affected by latest committed updates.

Committed in rev 10581 and distributed by mail.

**#24 - 07/30/2014 10:46 AM - Eric Faulhaber**

- Status changed from WIP to Closed

- % Done changed from 80 to 100

**#25 - 11/16/2016 12:06 PM - Greg Shah**

- Target version changed from Milestone 11 to Cleanup and Stabilization for Server Features

**Files**

---

om_upd20140711a.zip	356 KB	07/11/2014	Ovidiu Maxiniuc
om_upd20140714a.zip	356 KB	07/14/2014	Ovidiu Maxiniuc