

Database - Feature #2339

generate schema-triggers.xml from Java annotations at build time

07/14/2014 02:02 PM - Eric Faulhaber

Status: New	Start date:
Priority: Low	Due date:
Assignee:	% Done: 0%
Category:	Estimated time: 0.00 hour
Target version:	version:
billable: No	
vendor_id: GCD	
Description	
Related issues:	
Related to Database - Bug #2337: trigger scanning defect	Closed

History

#1 - 07/14/2014 02:10 PM - Eric Faulhaber

- Priority changed from Normal to Low

See discussion in [#2337](#), beginning at note 9, and particularly note 17.

The idea is to emit Java annotations related to schema triggers into the converted code, write a tool to extract these and generate schema-triggers.xml in src/{application path}/dmo, and enhance the application build to run this tool prior to (as a dependency of) the jar target.

We will have to be clever as to how we detect whether this needs to be done, so as to only increase the build time with this step when actually needed.

The current P2J bzd revision (10569) emits the annotations into converted code, but is about to be replaced by an update which generates schema-triggers.xml directly during conversion. We should refer to rev 10569 as a starting point when implementing this feature.

#2 - 07/14/2014 02:10 PM - Eric Faulhaber

- Project changed from Liberty to Database

#3 - 07/14/2014 02:18 PM - Ovidiu Maxiniuc

I was about to reply on 2337. I am not sure you understand the utility of schema-triggers.xml.

The xml file is created because it is needed to temporary store data between two unrelated places in the conversion pipeline (schema/p2o.xml to convert/base_structure.xml). In the second place the structure is populated with additional data only available at this point that adds the "cache" function to the file. When triggers are configured, their annotation is used, the xml only provides the name of the class to load instead of searching them in jar/classpath.

So, the file is there, we have it, I am saving it now to the correct location now (src/{application path}/dmo), I am a little bit delayed by computing the right path using the parameters from p2j.cfg.xml.

#4 - 07/14/2014 02:42 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

I was about to reply on 2337. I am not sure you understand the utility of schema-triggers.xml.

The xml file is created because it is needed to temporary store data between two unrelated places in the conversion pipeline (schema/p2o.xml to convert/base_structure.xml). In the second place the structure is populated with additional data only available at this point that adds the "cache" function to the file. When triggers are configured, their annotation is used, the xml only provides the name of the class to load instead of searching them in jar/classpath.

I understand schema-triggers.xml is an artifact of conversion, and an intermediate form of it is needed during conversion, and that you recently enhanced it to allow you to bypass the class scanning done at server startup. But is there anything in that file (which is needed at runtime), which could not be populated entirely from the contents of the SchemaTrigger annotations in the converted trigger procedures? If not, then I think the idea to generate the XML at build time is valid.

Also, is there anything in that file which we *don't* need at runtime (i.e., information which is *only* needed during conversion)? If so, we probably should have a different XML file which contains only what is needed at runtime.

So, the file is there, we have it, I am saving it now to the correct location now (src/{application path}/dmo), I am a little bit delayed by computing the right path using the parameters from p2j.cfg.xml.

I guess the best way without adding a dedicated entry to the p2j.cfg.xml file would be to derive the {application path} from the pkgroot setting (i.e., getConfigParameter("pkgroot") and replacing the dots with slashes).

#5 - 07/14/2014 02:45 PM - Eric Faulhaber

Eric Faulhaber wrote:

... and replacing the dots with slashes).

What I should have written was "... and replacing the dots with the value of getConfigParameter("file-separator")".

#6 - 07/14/2014 03:22 PM - Ovidiu Maxiniuc

Eric Faulhaber wrote:

I understand schema-triggers.xml is an artifact of conversion, and an intermediate form of it is needed during conversion, and that you recently enhanced it to allow you to bypass the class scanning done at server startup. But is there anything in that file (which is needed at runtime), which could not be populated entirely from the contents of the SchemaTrigger annotations in the converted trigger procedures? If not, then I think the idea to generate the XML at build time is valid.

Also, is there anything in that file which we *don't* need at runtime (i.e., information which is *only* needed during conversion)? If so, we probably should have a different XML file which contains only what is needed at runtime.

You are perfectly right for the 1st part. The xml initially held legacy informations (resolved trigger procedure file, event type, table name) they are used to create trigger annotations.

At the second step of processing, the initial infos are converted to java (respectively: trigger class, DatabaseEventType, associated Buffer class) and also written back. In base_structure all the knowlegde is there, in both java and legacy. At this moment, it is simply saved back, using the same xml document that was loaded.

At runtime 4 infos are needed for configuring the trigger: database, class-name, event and special case for assign triggers, the property. All informations can be extracted from class annotations. We only need, for each database, the list of the files to be loaded dynamically instead of searching them in classpath.

There are a few improvements we can do to this file before saving it:

- dropping trigger entries for triggers that could not be matched (and the warning was already issued)
- dropping all attributes except database and classname (initially, they are not needed after base_structure processing, the entire file was simply abandoned), we can alter the xml document to remove the information that was needed only at conversion time.
- save it with another name (trigger-cache.xml).

Creating another file does make sense, maybe with a slightly modified structure (intermediary node level for database ?).

I guess the best way without adding a dedicated entry to the p2j.cfg.xml file would be to derive the {application path} from the pkgroot setting (i.e., getConfigParameter("pkgroot") and replacing the dots with slashes).

Exactly, it seems to be working now. A few path separators file dependent also added:

```
location = sprintf("%s%s%s%sdmo%sschema-triggers.xml",
                  getConfigParameter("output-root"),
                  fileSep,
                  getConfigParameter("pkgroot").replaceAll("\\.", quoteReplacement(fileSep)),
                  fileSep,
                  fileSep)
```