

## Database - Feature #2235

### fix deployment issues related to dynamic database conversion resources

01/25/2014 08:09 PM - Eric Faulhaber

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Stanislav Lomany	<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>vendor_id:</b>	GCD
<b>billable:</b>	No		
<b>Description</b>			
<b>Related issues:</b>			
Related to Database - Feature #1651: add support for dynamic database features		<b>Closed</b>	<b>04/02/2013 07/16/2013</b>
Related to Database - Feature #4722: data import should be able to run with o...		<b>Test</b>	

## History

### #1 - 01/25/2014 08:15 PM - Eric Faulhaber

In [#1652](#), note 135, Stanislav wrote:

The following files and directories have to be accessed from the server classpath:

```
data
cfg
rules
name_map.cache
hibernate-mapping-3.0.dtd (if local dtd file is linked in hibernate template)
```

Currently, these resources -- required at runtime by the P2J server -- are made available via the file system in development environments, using symbolic links to the directories containing these resources, from the current directory. We need a better way to deploy these features, which:

- eliminates dependencies where easily possible;
- builds required dependencies the jars;
- loads these dependencies from the jars.

This will involve changes to both the P2J project (to load conversion rule-sets, for instance from the p2j.jar or a related jar), and to each customer project (to load other conversion artifacts from customer-specific jar files).

**#2 - 01/26/2014 09:56 AM - Eric Faulhaber**

- Assignee set to *Stanislav Lomany*

**#3 - 03/12/2014 08:22 AM - Stanislav Lomany**

Eric, after unnecessary dependencies will be eliminated, I assume that rules should be stored in p2j[rt].jar. data, cfg and name\_map.cache should be customer-specific, so it will be included in the customer's future build file. For <big\_chui\_regression\_application> we don't need these dependencies. How these file should be handled for testcases project - should I include them in p2j.jar?

**#4 - 03/12/2014 11:02 AM - Eric Faulhaber**

Stanislav Lomany wrote:

How these file should be handled for testcases project - should I include them in p2j.jar?

No, they should be included in the testcases.jar file that is built by the testcases project's build.xml.

**#5 - 03/20/2014 12:46 PM - Stanislav Lomany**

- File *svl\_upd20140320a.zip* added

Update for review (P2J part).

**#6 - 03/20/2014 01:34 PM - Stanislav Lomany**

- File *svl\_upd20140320b.zip* added

Update (testcases project part).

**#7 - 03/20/2014 02:29 PM - Eric Faulhaber**

Code review 20140320a:

- In build.xml, please restrict the rulesets copied into the jar to the minimum needed for the dynamic database conversion support, rather than copying rules/\*\*. Also, please remove the hard tabs introduced into the jar target in build.xml.
- Nice catch on the name\_map.cache loading in base\_structure.xml; I hope that saves some time. I suspect we don't ever need to load schema-triggers.xml at runtime for the dynamic temp-table and query work we do, so that probably can be removed in runtime query mode as well (Ovidiu: please comment if needed). Any unnecessary processing we can avoid to improve the performance of runtime conversion is important.
- Please elaborate on the changes to SchemaLoader.loadAst. Is the idea that we always go down the else path in runtime mode? Please add comments/javadoc to the code to explain this.
- It always makes me nervous when we suppress exceptions. What is the reason for this in SchemaConfig.getAndCheckSchemaFile?
- In DynamicQueryHelper, why are we loading permanent schemas first from file, then from jar? Shouldn't we just load from jar? Is this for development use? I always rebuild jars during development, maybe other people work differently.

**#8 - 03/20/2014 02:39 PM - Eric Faulhaber**

Code review 20140320b:

The content looks good, but please replace the hard tabs with soft tabs in build.xml before you check this in.

So, I notice you are not copying name\_map.cache into the testcases.jar (and you had related changes in 20140320a to not load it in runtime query mode). If we do not need to load this in runtime query mode, why have we needed the symlink to this file up until now?

**#9 - 03/20/2014 02:41 PM - Eric Faulhaber**

Eric Faulhaber wrote:

So, I notice you are not copying name\_map.cache into the testcases.jar (and you had related changes in 20140320a to not load it in runtime query mode). If we do not need to load this in runtime query mode, why have we needed the symlink to this file up until now?

Never mind, I think I just answered my own question: by removing the unnecessary loading in base\_structure.xml, you eliminated the unnecessary dependency on the resource, right?

**#10 - 03/20/2014 02:42 PM - Ovidiu Maxiniuc**

Eric Faulhaber wrote:

I suspect we don't ever need to load schema-triggers.xml at runtime for the dynamic temp-table and query work we do, so that probably can be removed in runtime query mode as well (Ovidiu: please comment if needed). Any unnecessary processing we can avoid to improve the performance of runtime conversion is important.

No, this file is only needed during the conversion. Information stored temporarily in there is available at runtime as annotations in schema-trigger classes. Ignore it.

LE: In fact, temp-tables cannot have schema triggers, in fact no database triggers at all. Seems logical.

**#11 - 03/20/2014 03:32 PM - Stanislav Lomany**

- In build.xml, please restrict the rulesets copied into the jar to the minimum needed for the dynamic database conversion support, rather than copying rules/\*\*.

Do you want me to pick specific *files* or specific rules/ *subdirectories* (like rules/annotations/\*\*)?

- It always makes me nervous when we suppress exceptions. What is the reason for this in SchemaConfig.getAndCheckSchemaFile?

Exception is thrown if the file cannot be found, and in this case this function returns null which is how it supposed to work.

- Please elaborate on the changes to SchemaLoader.loadAst. Is the idea that we always go down the else path in runtime mode? Please add comments/javadoc to the code to explain this.
- In DynamicQueryHelper, why are we loading permanent schemas first from file, then from jar? Shouldn't we just load from jar? Is this for development use? I always rebuild jars during development, maybe other people work differently.

Yes, I made that for development use. But we may assume that in conversion mode we read configuration from files and in runtime - from jars (or resources directory). In this case those who don't use jars in development (like me) will have to update their IDE projects.

schema-triggers.xml

I'll disable its loading at runtime.

by removing the unnecessary loading in base\_structure.xml, you eliminated the unnecessary dependency on the resource, right?

Right.

**#12 - 03/20/2014 03:59 PM - Eric Faulhaber**

Stanislav Lomany wrote:

- In build.xml, please restrict the rulesets copied into the jar to the minimum needed for the dynamic database conversion support, rather than copying rules/\*\*.

Do you want me to pick specific *files* or specific rules/ *subdirectories* (like rules/annotations/\*\*)?

Ideally, we would like to include only the specific files needed. Although this probably could be scoped to its own task about performance, I would like you to create some more focused, top-level rule-sets to replace those of broader scope than what we need for dynamic database (like annotations/annotations.xml, possibly convert/core-conversion.xml). The less we load per PatternEngine instance to do this specific runtime job, the better.

Let me know if you encounter problems with this approach, or if you feel the level of effort is more than a day or so.

**#13 - 03/20/2014 04:08 PM - Eric Faulhaber**

Eric Faulhaber wrote:

The less we load per PatternEngine instance to do this specific runtime job, the better.

Actually, I guess the separate, top-level rule-sets are unnecessary, as we already are limiting this with the load-condition="runtime-query". This should give you a good starting indicator as to which individual files we need in the jar. There will be some additional include dependencies to work out, but many of them should be shared.

**#14 - 03/24/2014 04:58 AM - Stanislav Lomany**

- File svl\_upd20140324a.zip added

Copies only specific rules. Removed tabs. Removed loading of schema-triggers.xml in runtime mode.  
Would you like me to leave the code as is or to use files/resources only in corresponding conversion/runtime modes?

**#15 - 03/24/2014 04:59 AM - Stanislav Lomany**

- File svl\_upd20140324b.zip added

Removed tabs.

**#16 - 03/24/2014 11:46 AM - Eric Faulhaber**

Code review 20140324a/b:

Changes look good. I would like you to still please add a comment to SchemaLoader.loadAst explaining the logic with respect to where we expect to find the resource under different circumstances.

Would you like me to leave the code as is or to use files/resources only in corresponding conversion/runtime modes?

You can leave it as is if you find it helpful for development, but that means that anyone making changes to the list of resources in use will have to

remember to test that they are properly loaded from the jar file, not just from the file system.

**#17 - 03/24/2014 12:27 PM - Stanislav Lomany**

Eric, I recently met the rules loading code and it uses "non-development" approach, so, considering this and overall arguments, I think I'll also go "non-development" way.

**#18 - 03/25/2014 10:40 AM - Stanislav Lomany**

- File *svl\_upd20140325a.zip* added

Removed "development" mode.

**#19 - 03/25/2014 11:20 AM - Eric Faulhaber**

Code review 20140325a:

All looks good except the change to XmlFilePlugin. This class is meant to be for generic AST processing and should remain unaware of P2J specifics, but you have injected awareness of the P2J Configuration class. Instead, please add a method to load a tree from a resource and use Configuration outside this class (i.e., in DynamicQueryHelper) to make the decision as to which load method to use.

Please make this change and regression test the update. I do not need to review again if this is the only change you make.

**#20 - 03/25/2014 03:05 PM - Stanislav Lomany**

All looks good except the change to XmlFilePlugin. This class is meant to be for generic AST processing and should remain unaware of P2J specifics, but you have injected awareness of the P2J Configuration class. Instead, please add a method to load a tree from a resource and use Configuration outside this class (i.e., in DynamicQueryHelper) to make the decision as to which load method to use.

I'll make XmlFilePlugin constructor take optional boolean parameter which should be true if the plugin instance is supposed to handle resources rather than files.

**#21 - 03/28/2014 07:00 AM - Stanislav Lomany**

- File *svl\_upd20140326a.zip* added

Committed to bzt rev 10498.

**#22 - 03/28/2014 07:02 AM - Stanislav Lomany**

- File *deleted (svl\_upd20140320b.zip)*

**#23 - 03/28/2014 07:02 AM - Stanislav Lomany**

- File *deleted (svl\_upd20140324a.zip)*

**#24 - 03/28/2014 07:02 AM - Stanislav Lomany**

- File deleted (svl\_upd20140325a.zip)

**#25 - 03/28/2014 07:03 AM - Stanislav Lomany**

- File deleted (svl\_upd20140320a.zip)

**#26 - 03/28/2014 09:27 AM - Greg Shah**

Please don't remove archives unless they have been uploaded mistakenly (and have a proper replacement). For example, the 0325a zip was a legitimate version of the code and there was a code review. Someone coming along later can no longer see the code to which the review refers. That is a problem.

**#27 - 03/28/2014 04:12 PM - Eric Faulhaber**

One thing I just realized will be an issue: it is possible to run the static conversion of a project on one platform (say, Linux) and then run the converted result on another (say, Windows). This will require that we load the correct version of `cfg/p2j.cfg.xml` for runtime use. The changes to Configuration in `svl_upd20140326a.zip` only load the default configuration file. Please update the new "load from resource" logic to detect the current platform and load the appropriate, platform-specific version of the file, instead of the default version.

There is no need to do runtime regression testing for this change -- just do a conversion regression test and manually test that runtime conversion for a simple dynamic test case still works on Linux. Do you have a development/test environment running on Windows where you can test this (using the `testcases.jar` generated on Linux)? If not, we can ask someone else who does to run your dynamic test case on Windows.

**#28 - 03/28/2014 11:43 PM - Eric Faulhaber**

I tried to run with this update applied and I found a regression. In `DynamicQueryHelper.loadPermanentSchemas`, you removed the loading of the AST registry (which needs to be a resource in the application jar file). This breaks any dynamic queries which reference static (i.e., non dynamically-defined) schema elements, as we get an NPE when `InMemoryRegistryPlugin.loadTree` tries to find and clone such an AST:

```

Caused by: java.lang.NullPointerException
    at com.goldencode.ast.InMemoryRegistryPlugin.loadTree(InMemoryRegistryPlugin.java:49)
    at com.goldencode.ast.AstManager.loadTree(AstManager.java:221)
    at com.goldencode.p2j.schema.P2OLookup.loadAst(P2OLookup.java:1069)
    at com.goldencode.p2j.schema.P2OLookup.initialize(P2OLookup.java:817)
    at com.goldencode.p2j.schema.P2OLookup.<init>(P2OLookup.java:217)
    at com.goldencode.p2j.schema.P2OLookup.getInstance(P2OLookup.java:681)
    at com.goldencode.p2j.schema.P2OLookup.javaName(P2OLookup.java:445)
    at com.goldencode.p2j.schema.P2OLookup.javaPropertyName(P2OLookup.java:511)
    at com.goldencode.p2j.schema.P2OAccessWorker$Library.javaPropertyName(P2OAccessWorker.java:212)
    at com.goldencode.expr.CE70821.execute(Unknown Source)
    at com.goldencode.expr.Expression.execute(Expression.java:356)
    at com.goldencode.p2j.pattern.Rule.apply(Rule.java:401)
    at com.goldencode.p2j.pattern.Rule.executeActions(Rule.java:640)
    at com.goldencode.p2j.pattern.Rule.coreProcessing(Rule.java:609)
    at com.goldencode.p2j.pattern.Rule.apply(Rule.java:440)
    at com.goldencode.p2j.pattern.RuleContainer.apply(RuleContainer.java:531)
    at com.goldencode.p2j.pattern.RuleSet.apply(RuleSet.java:50)
    at com.goldencode.p2j.pattern.AstWalker.walk(AstWalker.java:213)
    at com.goldencode.p2j.pattern.AstWalker.walk(AstWalker.java:160)
    at com.goldencode.p2j.pattern.PatternEngine.apply(PatternEngine.java:1391)
    at com.goldencode.p2j.pattern.PatternEngine.processAst(PatternEngine.java:1289)
    at com.goldencode.p2j.pattern.PatternEngine.run(PatternEngine.java:888)
    at com.goldencode.p2j.persist.DynamicQueryHelper.runPattern(DynamicQueryHelper.java:1053)
    at com.goldencode.p2j.persist.DynamicQueryHelper.parse(DynamicQueryHelper.java:232)
    at com.goldencode.p2j.persist.DynamicQueryHelper.parseQuery(DynamicQueryHelper.java:364)
    at com.goldencode.p2j.persist.QueryWrapper.prepare(QueryWrapper.java:2854)
    ...

```

I tried to fix this quickly by adding `cfg/registry.xml` to the application jar and partially reverting the change to `DynamicQueryHelper.loadPermanentSchemas` to load the registry from the jar, but I found that `XMLPlugin(String, boolean)` has the following TODO:

```
if (useResources)
{
    // TODO implement loading registry from the resource
}
```

Please fix this.

**#29 - 03/29/2014 04:09 PM - Stanislav Lomany**

- Status changed from New to WIP

- File svl\_upd20140329a.zip added

Fix for review for issue in note 28.

**#30 - 03/29/2014 05:12 PM - Stanislav Lomany**

About registry.xml: Configuration class keeps absolute paths to configuration files, so if we want to load additional configuration xml files from resources, that requires a bit of work, so I prefer to wait with this functionality until it will be clear that we need these files.

About p2j.cfg.xml.{platform\_identifier}: I'm just curious - at what point we are referencing platform-specific files? I've looked in java and build code and could not find a reference.

**#31 - 03/29/2014 06:00 PM - Eric Faulhaber**

Code review 20140329a:

Looks good and fixes the issue. Please run conversion regression only and commit if it passes.

**#32 - 03/29/2014 06:06 PM - Eric Faulhaber**

Stanislav Lomany wrote:

About p2j.cfg.xml.{platform\_identifier}: I'm just curious - at what point we are referencing platform-specific files? I've looked in java and build code and could not find a reference.

That is a good point. The differences are primarily about the file and path separators and some other settings which are not relevant. For the limited conversion we are doing at runtime, I think we are just using this resource to find the location of the registry and the rule-sets, and that is set the same, regardless of platform. So, it probably does not matter which version is loaded.

**#33 - 03/29/2014 06:55 PM - Stanislav Lomany**



Just for the case I've changed file separator to \ and run my usual testcase - works OK, so most likely it really doesn't matter.

**#34 - 03/30/2014 07:23 AM - Stanislav Lomany**

- Status changed from WIP to Review

svl\_upd20140329a.zip was committed to bzs rev 10501.

**#35 - 03/30/2014 02:52 PM - Eric Faulhaber**

- % Done changed from 0 to 100

- Status changed from Review to Closed

**#36 - 11/16/2016 12:07 PM - Greg Shah**

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

**#37 - 07/01/2020 01:20 PM - Greg Shah**

- Related to Feature #4722: data import should be able to run with only the converted application jar file (and FWD) added

**Files**

---

svl_upd20140324b.zip	4.17 KB	03/24/2014	Stanislav Lomany
svl_upd20140326a.zip	94.5 KB	03/28/2014	Stanislav Lomany
svl_upd20140329a.zip	11.3 KB	03/29/2014	Stanislav Lomany