

## Database - Feature #5130

### improve management of p2jpl.jar in deployment scripts

02/04/2021 12:40 PM - Greg Shah

<b>Status:</b>	Rejected	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>vendor_id:</b>	GCD
<b>billable:</b>	No		
<b>Description</b>			

#### History

##### #1 - 02/04/2021 12:51 PM - Greg Shah

Our current devops scripting is implemented outside of FWD. We use Apache ant to script a range of conversion project tasks including the import of the database. The p2jpl.jar is an important dependency whose management is completely manual at this point. This leads to problems when we change the p2jpl.jar in FWD but forget to drop and reinstall it in a runtime database.

I've discussed this with Eric, we think the best approach (for now) is:

- Add a target to copy the p2jpl.jar to /usr/share/java/p2jpl.jar (or wherever it is needed on Windows) and then drop and reinstall p2jpl.jar in the database.
  - This will need to use sudo similar to how we implement the installation of the spawner.
  - The location should be customizable in case the default is not appropriate for the customer's installation.
  - It should only be done if the database target is PostgreSQL.
- Add a dependency to deploy.prepare or possibly something else? We don't like that deploy.prepare would have this dependency, but it is not clear if there is a better location.
- Add a version UDF that is customized at build time similar to how we version FWD.
- Call the version UDF during server startup (only for PostgreSQL) and report if there is a mis-match of versions. Don't abend, just log it.

##### #2 - 02/04/2021 01:05 PM - Igor Skorniyakov

I use the following "quick and dirty" approach based on direnv utility in the situations when additional steps are required after some common operation like build.

1. define an alias alias \_R='eval \$R' on the .bashrc file.
2. and definition of the R environment variable to the .envrc file. For example the .envrc in the FWD project root in my environment is:

```
export R='./gradlew core; sudo cp build/lib/p2jpl.jar /usr/share/java; sudo chmod a+x /usr/share/java/p2jpl.jar'
```

3. use \_R command to build the project with additional post steps.

### #3 - 02/04/2021 01:09 PM - Greg Shah

Clever!

### #4 - 02/04/2021 03:19 PM - Eric Faulhaber

Greg Shah wrote:

Add a version UDF that is customized at build time similar to how we version FWD.

This already exists: `Functions.getFWDVersion()`. To use it from SQL:

```
select getfwdversion();
```

For example:

```
p2j_test=# select getfwdversion();
          getfwdversion
-----
FWD v4.0.0_p2j_3821c_11521
(1 row)
```

### #5 - 02/08/2021 03:49 PM - Eric Faulhaber

Does anyone know why we have this in the "template" `build_db.xml` (not really a template, but it tends to get copied across projects)?

```
<condition property="p2jpl.jar.path" value="${deploy.home.abs}/lib/p2jpl.jar" else="/usr/share/java/p2jpl.jar">
  <isset property="isWindows" />
</condition>
```

This is used to set up the `p2jpl.jar` installation during database import.

Why do we use the most current `p2jpl.jar` version from `${deploy.home.abs}/lib/` only in the Windows case, while we use the most likely out-of-date version from `/usr/share/java/` for non-Windows? It seems this was a deliberate choice, otherwise it would always be `${deploy.home.abs}/lib/p2jpl.jar`. A permissions issue, perhaps? Copying from `${deploy.home.abs}/lib/p2jpl.jar` seems like it would work on Linux for a developer build, but maybe not for certain customer environments?

**#6 - 02/08/2021 04:28 PM - Constantin Asofiei**

Eric Faulhaber wrote:

Why do we use the most current p2jpl.jar version from `${deploy.home.abs}/lib/` only in the Windows case, while we use the most likely out-of-date version from `/usr/share/java/` for non-Windows? It seems this was a deliberate choice, otherwise it would always be `${deploy.home.abs}/lib/p2jpl.jar`. A permissions issue, perhaps?

Correct, is a permission issue. The postgres user will need to be able to read this file.

**#7 - 06/09/2023 09:11 AM - Greg Shah**

- % Done changed from 0 to 100

- Status changed from New to Rejected

I'm closing this as unneeded. We are moving all customers to the native UDFs and PL/Java is deprecated at this time. I see no reason to invest time in better management of PL/Java.