CONFIGURE JBOSS / WILDFLY DATASOURCE WITH MAVEN

http://www.radcortez.com/configure-jboss-wildfly-datasource-with-maven/

Most Java EE applications use database access in their business logic, so developers are often faced with the need to configure drivers and database connection properties in the application server. In this post, we are going to automate that task for JBoss / [Wildfly](http://wildfly.org/)and a Postgre database using [Maven](http://maven.apache.org/). The work is based on my World of Warcraft Auctions Batch application from the previous [post](http://www.radcortez.com/java-ee-7-batch-processing-and-world-of-warcraft-part-1/).

Maven Configuration

Let’s start by adding the following to our pom.xml:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | <plugin>      <groupId>org.wildfly.plugins</groupId>      <artifactId>wildfly-maven-plugin</artifactId>      <version>1.0.2.Final</version>      <configuration>          <executeCommands>              <batch>false</batch>              <scripts>                  <script>target/scripts/${cli.file}</script>              </scripts>          </executeCommands>      </configuration>      <dependencies>          <dependency>              <groupId>org.postgresql</groupId>              <artifactId>postgresql</artifactId>              <version>9.3-1102-jdbc41</version>          </dependency>      </dependencies>  </plugin> |

We are going to use the [Wildfly Maven Plugin](https://docs.jboss.org/wildfly/plugins/maven/latest/) to execute scripts with commands in the application server. Note that we also added a dependency to the Postgre driver. This is for [Maven](http://maven.apache.org/) to download the dependency, because we are going to need it later to add it to the server. There is also a ${cli.file} property that is going to be assigned to a profile. This is to indicate which script we want to execute.

Let’s also add the following to the pom.xml:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | <plugin>      <groupId>org.apache.maven.plugins</groupId>      <artifactId>maven-resources-plugin</artifactId>      <version>2.6</version>      <executions>          <execution>              <id>copy-resources</id>              <phase>process-resources</phase>              <goals>                  <goal>copy-resources</goal>              </goals>              <configuration>                  <outputDirectory>${basedir}/target/scripts</outputDirectory>                  <resources>                      <resource>                          <directory>src/main/resources/scripts</directory>                          <filtering>true</filtering>                      </resource>                  </resources>                  <filters>                      <filter>${basedir}/src/main/resources/configuration.properties</filter>                  </filters>              </configuration>          </execution>      </executions>  </plugin> |

With the [Resources Maven Plugin](http://maven.apache.org/plugins/maven-resources-plugin/) we are going to filter the script files contained in thesrc/main/resources/scripts and replace them with the properties contained in${basedir}/src/main/resources/configuration.properties file.

Finally lets add a few Maven profiles to the pom.xml, with the scripts that we want to run:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | <profiles>      <profile>          <id>install-driver</id>          <properties>              <cli.file>wildfly-install-postgre-driver.cli</cli.file>          </properties>      </profile>        <profile>          <id>remove-driver</id>          <properties>              <cli.file>wildfly-remove-postgre-driver.cli</cli.file>          </properties>      </profile>        <profile>          <id>install-wow-auctions</id>          <properties>              <cli.file>wow-auctions-install.cli</cli.file>          </properties>      </profile>        <profile>          <id>remove-wow-auctions</id>          <properties>              <cli.file>wow-auctions-remove.cli</cli.file>          </properties>      </profile>  </profiles> |

Wildfly Script Files

Add Driver

The scripts with the commands to add a Driver:

**wildfly-install-postgre-driver.cli**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | # Connect to Wildfly instance  connect    # Create Oracle JDBC Driver Module  # If the module already exists, Wildfly will output a message saying that the module already exists and the script exits.  module add \      --name=org.postgre \      --resources=${settings.localRepository}/org/postgresql/postgresql/9.3-1102-jdbc41/postgresql-9.3-1102-jdbc41.jar \      --dependencies=javax.api,javax.transaction.api    # Add Driver Properties  /subsystem=datasources/jdbc-driver=postgre: \      add( \          driver-name="postgre", \          driver-module-name="org.postgre") |

Database drivers are added to [Wildfly](http://wildfly.org/) as a module. In this was, the driver is widely available to all the applications deployed in the server. With ${settings.localRepository}we are pointing into the database driver jar downloaded to your local Maven repository. Remember the dependency that we added into the [Wildfly Maven Plugin](https://docs.jboss.org/wildfly/plugins/maven/latest/)? It’s to download the driver when you run the plugin and add it to the server. Now, to run the script we execute (you need to have the application server running):

mvn process-resources wildfly:execute-commands -P "install-driver"

The process-resources lifecycle is needed to replace the properties in the script file. In my case ${settings.localRepository} is replaced by /Users/radcortez/.m3/repository/. Check the target/scripts folder. After running the command, you should see the following output in the Maven log:

|  |  |
| --- | --- |
|  | {"outcome" => "success"} |

And on the server:

|  |  |
| --- | --- |
|  | INFO  [org.jboss.as.connector.subsystems.datasources] (management-handler-thread - 4) JBAS010404: Deploying non-JDBC-compliant driver class org.postgresql.Driver (version 9.3)  INFO  [org.jboss.as.connector.deployers.jdbc] (MSC service thread 1-4) JBAS010417: Started Driver service with driver-name = postgre |

**wildfly-remove-postgre-driver.cli**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | # Connect to Wildfly instance  connect    if (outcome == success) of /subsystem=datasources/jdbc-driver=postgre:read-attribute(name=driver-name)        # Remove Driver      /subsystem=datasources/jdbc-driver=postgre:remove    end-if    # Remove Oracle JDBC Driver Module  module remove --name=org.postgre |

This script is to remove the driver from the application server. Executemvn wildfly:execute-commands -P "remove-driver". You don’t need process-resources if you already executed the command before, unless you change the scripts.

Add Datasource

**wow-auctions-install.cli**  
The scripts with the commands to add a Datasource:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | # Connect to Wildfly instance  connect    # Create Datasource  /subsystem=datasources/data-source=WowAuctionsDS: \      add( \       jndi-name="${datasource.jndi}", \       driver-name=postgre, \       connection-url="${datasource.connection}", \       user-name="${datasource.user}", \       password="${datasource.password}")    /subsystem=ee/service=default-bindings:write-attribute(name="datasource", value="${datasource.jndi}") |

We also need a a file to define the properties:

**configuration.properties**

|  |  |
| --- | --- |
| 1  2  3  4 | datasource.jndi=java:/datasources/WowAuctionsDS  datasource.connection=jdbc:postgresql://localhost:5432/wowauctions  datasource.user=wowauctions  datasource.password=wowauctions |

Default Java EE 7 Datasource

Java EE 7, specifies that the container should provide a default Datasource. Instead of defining a Datasource with the JNDI name java:/datasources/WowAuctionsDS in the application, we are going to point our newly created datasource to the default one with/subsystem=ee/service=default-bindings:write-attribute(name="datasource", value="${datasource.jndi}"). In this way, we don’t need to change anything in the application. Execute the script with mvn wildfly:execute-commands -P "install-wow-auctions". You should get the following Maven output:

|  |
| --- |
| org.jboss.as.cli.impl.CommandContextImpl printLine  INFO: {"outcome" => "success"}  {"outcome" => "success"}  org.jboss.as.cli.impl.CommandContextImpl printLine  INFO: {"outcome" => "success"}  {"outcome" => "success"} |

And on the server:

|  |
| --- |
| INFO  [org.jboss.as.connector.subsystems.datasources] (MSC service thread 1-1) JBAS010400: Bound data source [java:/datasources/WowAuctionsDS] |

**wow-auctions-remove.cli**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | # Connect to Wildfly instance  connect    # Remove Datasources  /subsystem=datasources/data-source=WowAuctionsDS:remove    /subsystem=ee/service=default-bindings:write-attribute(name="datasource", value="java:jboss/datasources/ExampleDS") |

This is the script to remove the Datasource and revert the Java EE 7 default Datasource. Run it by executing mvn wildfly:execute-commands -P "remove-wow-auctions"

Conclusion

This post demonstrated how to automate add / remove Drivers to Wildfly instances and also add / remove Datasources. This is useful if you want to switch between databases or if you’re configuring a server from the ground up. Think about CI environments. These scripts are also easily adjustable to other drivers.

You can get the code from the [WoW Auctions Github repo](https://github.com/radcortez/wow-auctions), which uses this setup. Enjoy!