JDG and EAP Lab Setup Guide

Follow this guide to setup the lab environment for JDG labs

NOTE: If you are looking at the PDF version and have problems with for example copying text, the original Markdown is available here (https://github.com/tqvarnst/jdg-labs/blob/master/projects/lab-guides/lab-setup-guide.md) .

Prerequisits

- Host operating system should be Linux or a BSD based os like Mac OS X.
    - Alternative one of the above specified OS could be running in a virtual machine with a Desktop Manager
- The following software should be installed prior to the lab
    - OpenJDK or Oracle JDG using 1.6 or 1.7
    - Apache Maven (installed and on the path)
    - JBoss Developer Studio 7.X
- Download the following JBoss software as zip files (used later in the labs)
    - `jboss-datagrid-6.3.0-eap-modules-hotrod-java-client.zip`
    - `jboss-datagrid-6.3.0-eap-modules-library.zip`
    - `jboss-datagrid-6.3.0-maven-repository.zip`
    - `jboss-datagrid-6.3.0-quickstarts.zip`
    - `jboss-datagrid-6.3.0-server.zip`
    - `jboss-eap-6.2.4-full-maven-repository.zip`
    - `jboss-eap-6.3.0-maven-repository.zip`
    - `jboss-eap-6.3.0-quickstarts.zip`
    - `jboss-eap-6.3.0.zip`
- Download jdg-labs.zip (todo: provide link)

Setup Development environment

1. Unzip jdg-labs.zip in a suiteable directory (for example $HOME)

   ```
   $ unzip jdg-labs.zip -d $HOME
   ```

1. Change directory into the lab root dir

   ```
   $ cd ~/jdg-workshops/
   ```

2. Copy (or move) the downloaded jboss software into installs directory

3. Setup the development environment with the provided script
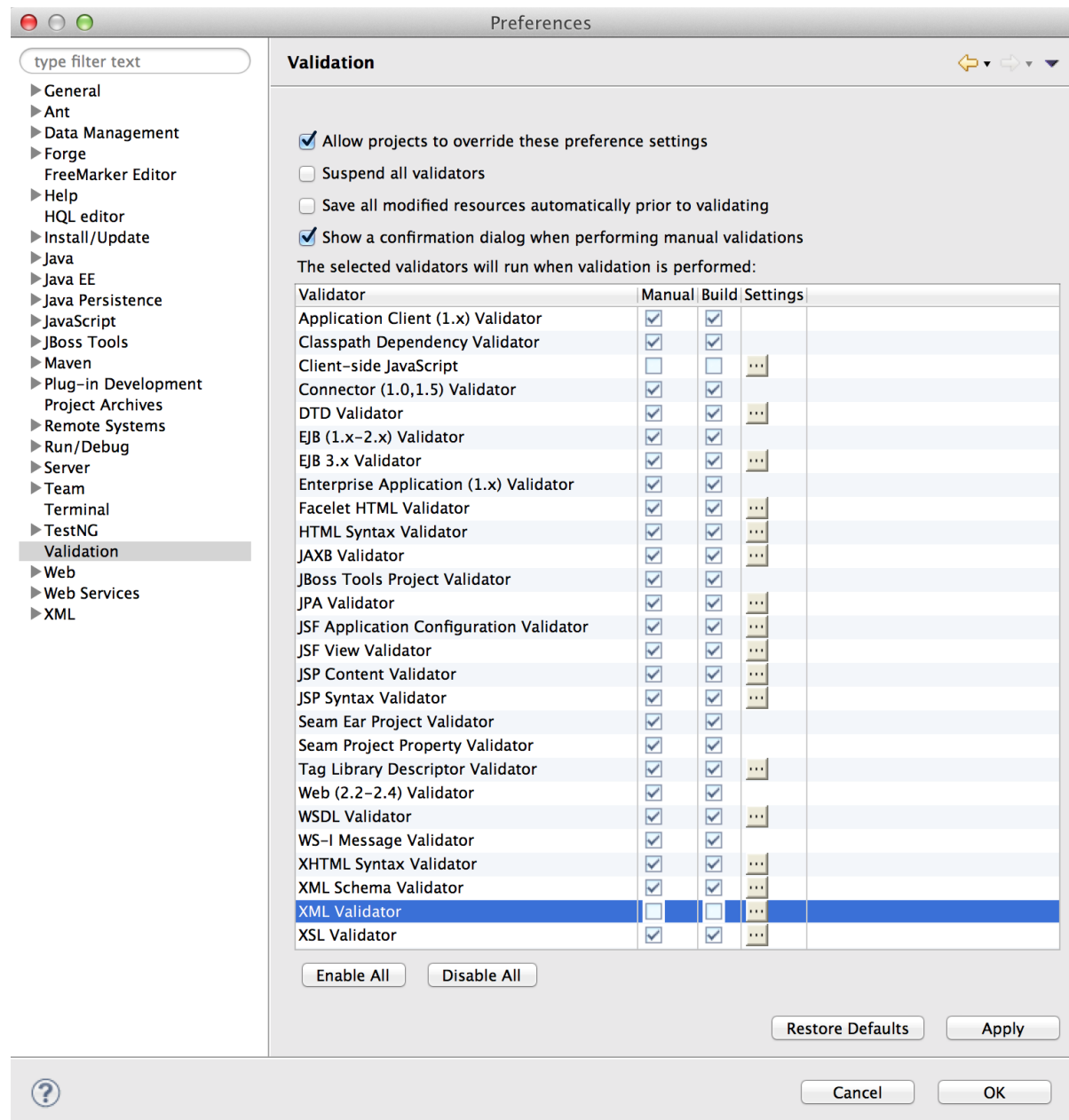
   ```
   $ sh init-dev.sh
   ```

4. Copy generated settings.xml to $HOME/.m2/ (don't forgett to backup any existins settings.xml)

   ```
   $ cp target/settings.xml ~/.m2/
   ```
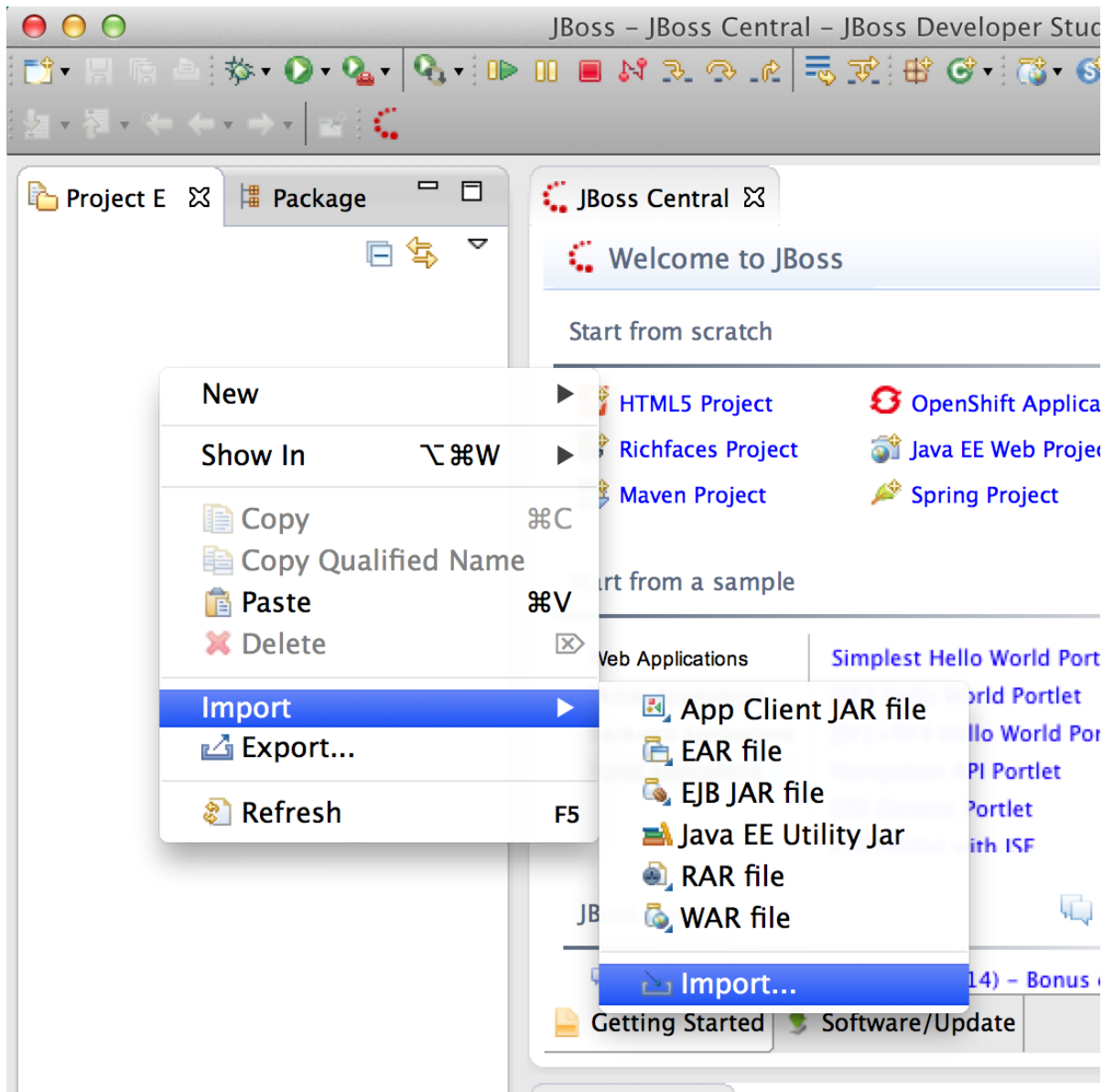
5. Start JBoss Developer Studio
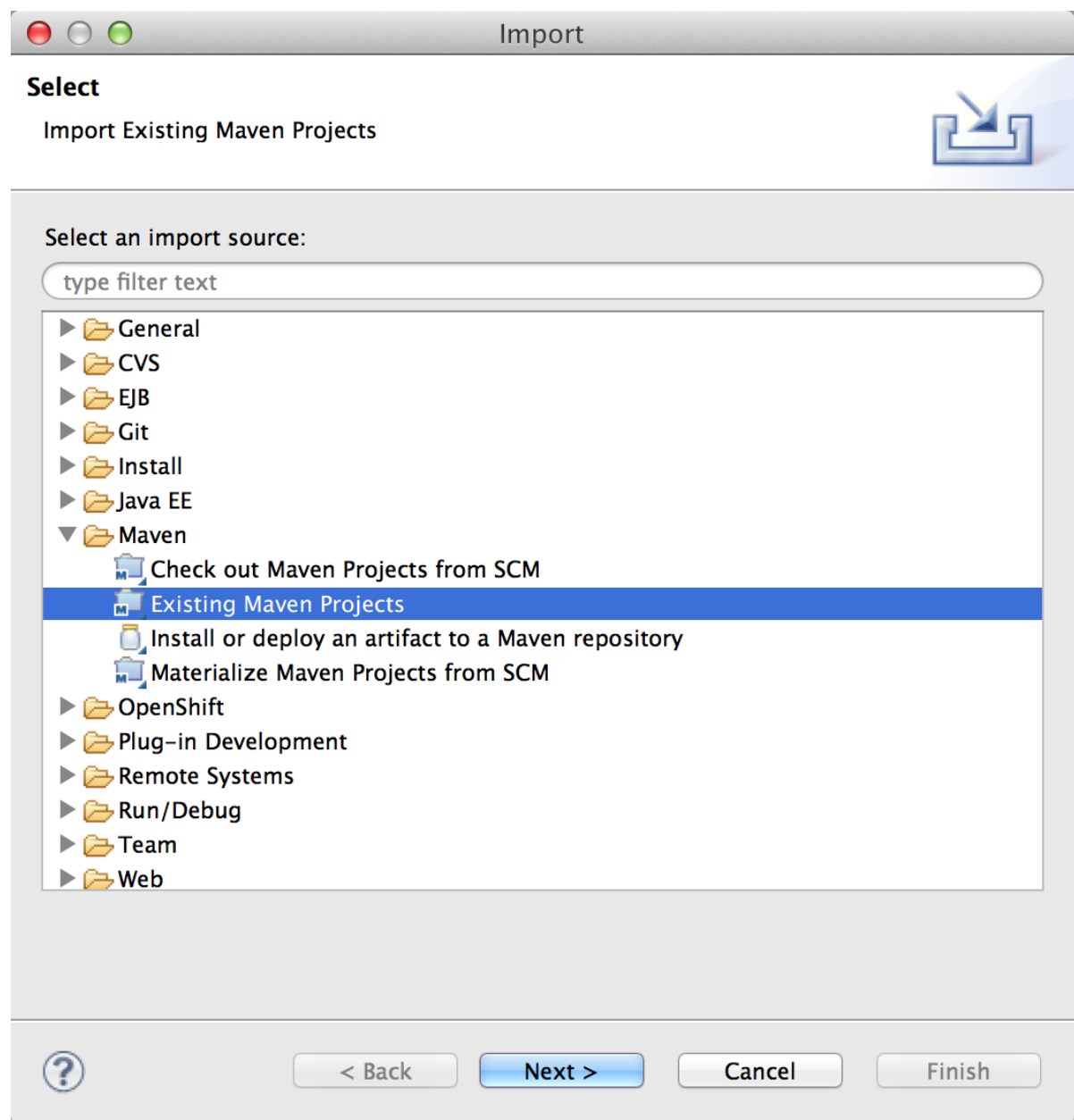
6. Turn off XML validation

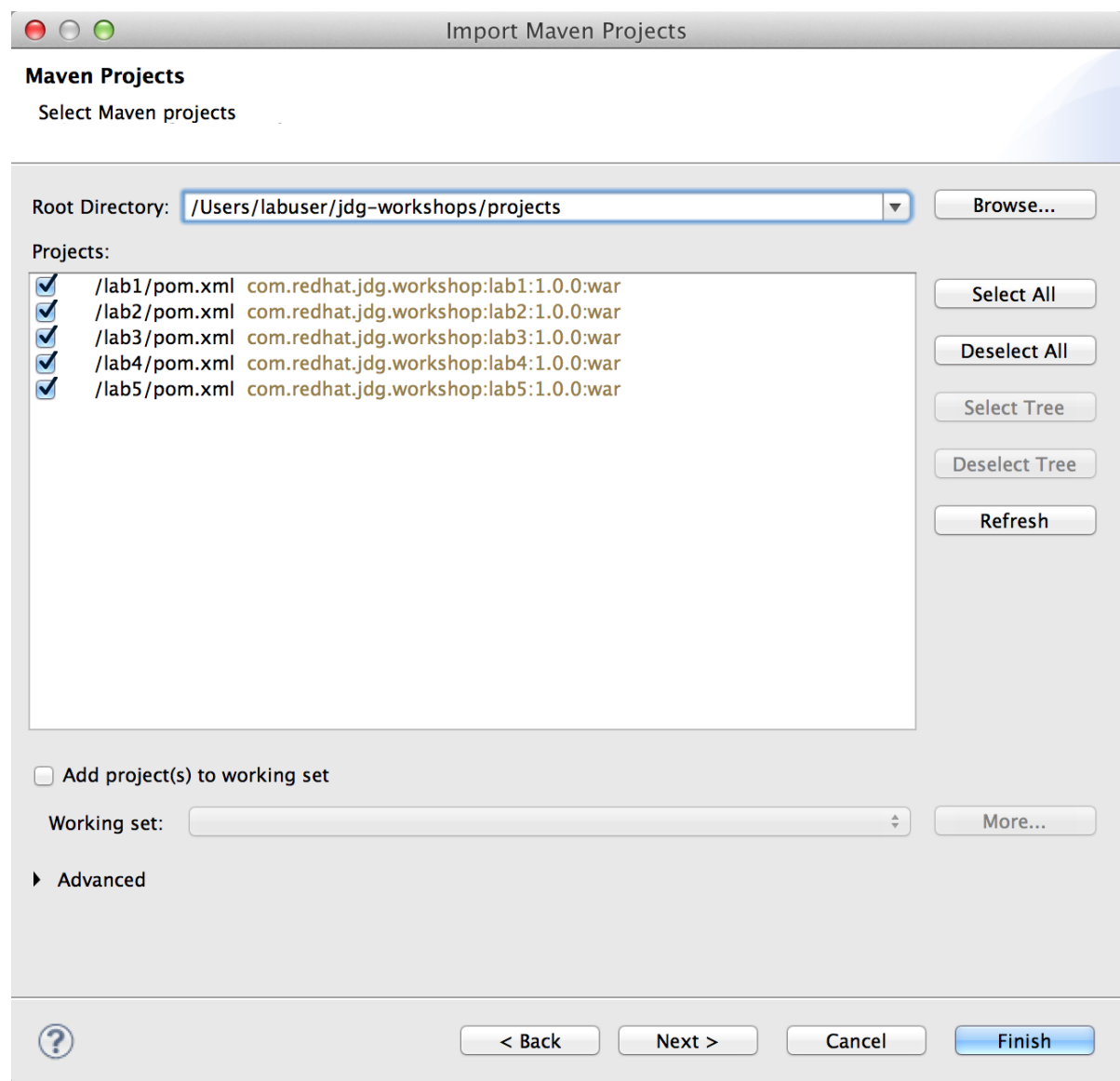   Preferences -> Validation

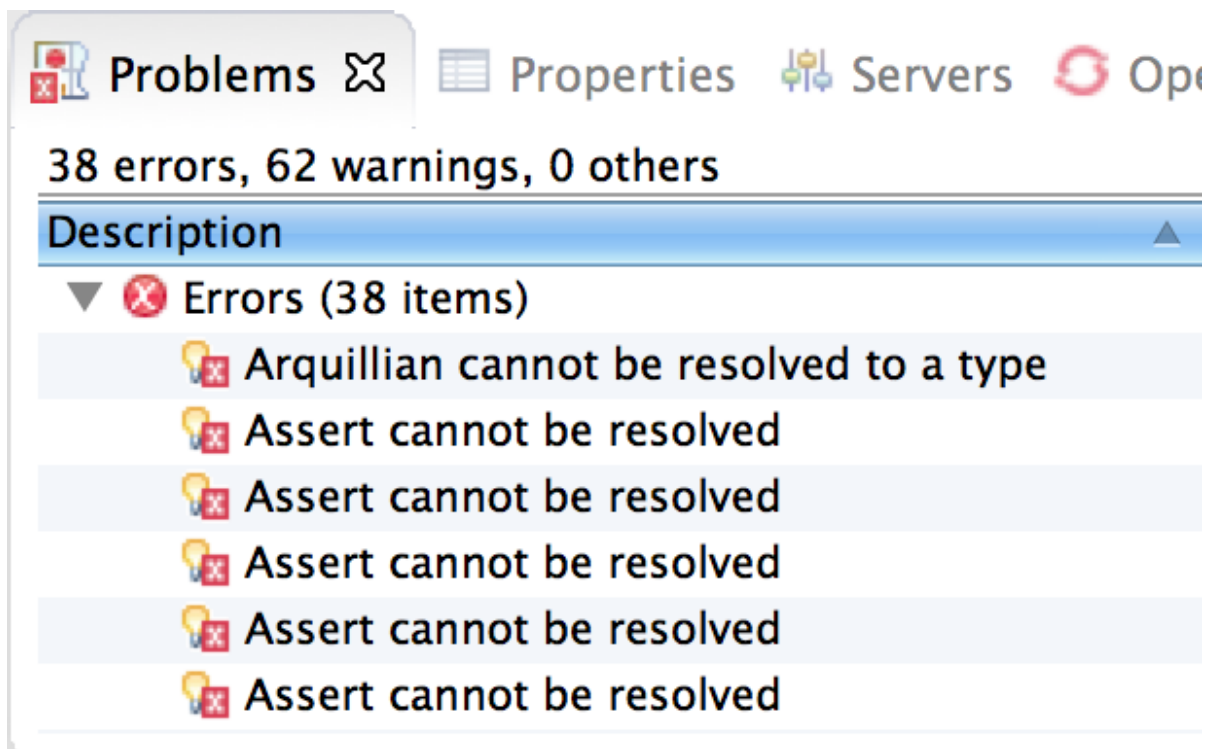7. Import the projects. Right click on the project area select Import -> Import ...

Select Existing Maven Project and click Next

Browse to the project directory (~/jdg−workshops/project) and select all lab projects

Window title: Import Maven Projects

Maven Projects

Select Maven projects

Root Directory: /Users/labuser/jdg-workshops/projects ▾   Browse...

Projects:

☑ /lab1/pom.xml com.redhat.jdg.workshop:lab1:1.0.0:war
☑ /lab2/pom.xml com.redhat.jdg.workshop:lab2:1.0.0:war
☑ /lab3/pom.xml com.redhat.jdg.workshop:lab3:1.0.0:war
☑ /lab4/pom.xml com.redhat.jdg.workshop:lab4:1.0.0:war
☑ /lab5/pom.xml com.redhat.jdg.workshop:lab5:1.0.0:war

Select All
Deselect All
Select Tree
Deselect Tree
Refresh

☐ Add project(s) to working set

Working set: ▴▾   More...

▸ Advanced

< Back   Next >   Cancel   Finish

8. At this point the workbench should have some compilation errors that looks something like this:

Problems ✕ | **Properties** | **Servers** | Op

38 errors, 62 warnings, 0 others

Description ▲

▼ ❌ Errors (38 items)

🔲 Arquillian cannot be resolved to a type

🔲 Assert cannot be resolved

🔲 Assert cannot be resolved

🔲 Assert cannot be resolved

🔲 Assert cannot be resolved

🔲 Assert cannot be resolved

9. To fix this you need to enable a Maven profile for Arquillian by right clicking on each project and select Maven -> Select Maven Profile...
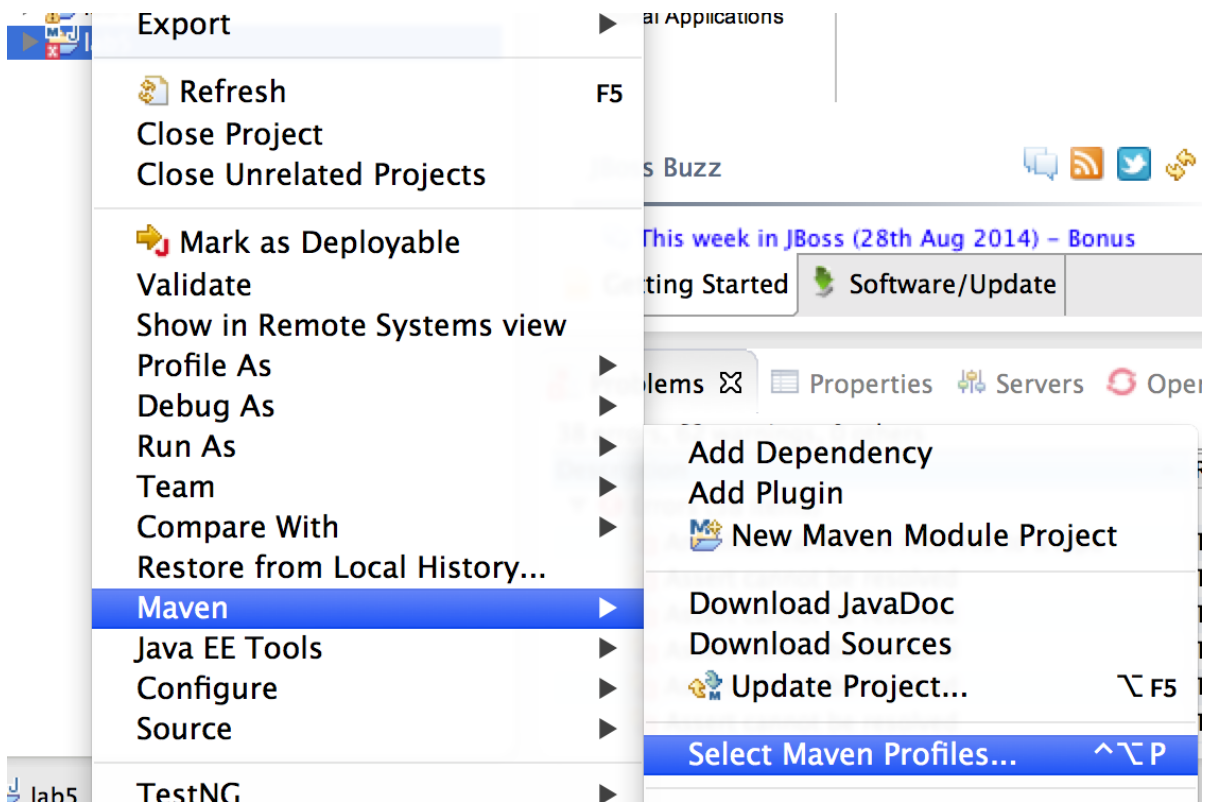
10. And in the dialog check the following maven profiles as illustrated by the picture below.
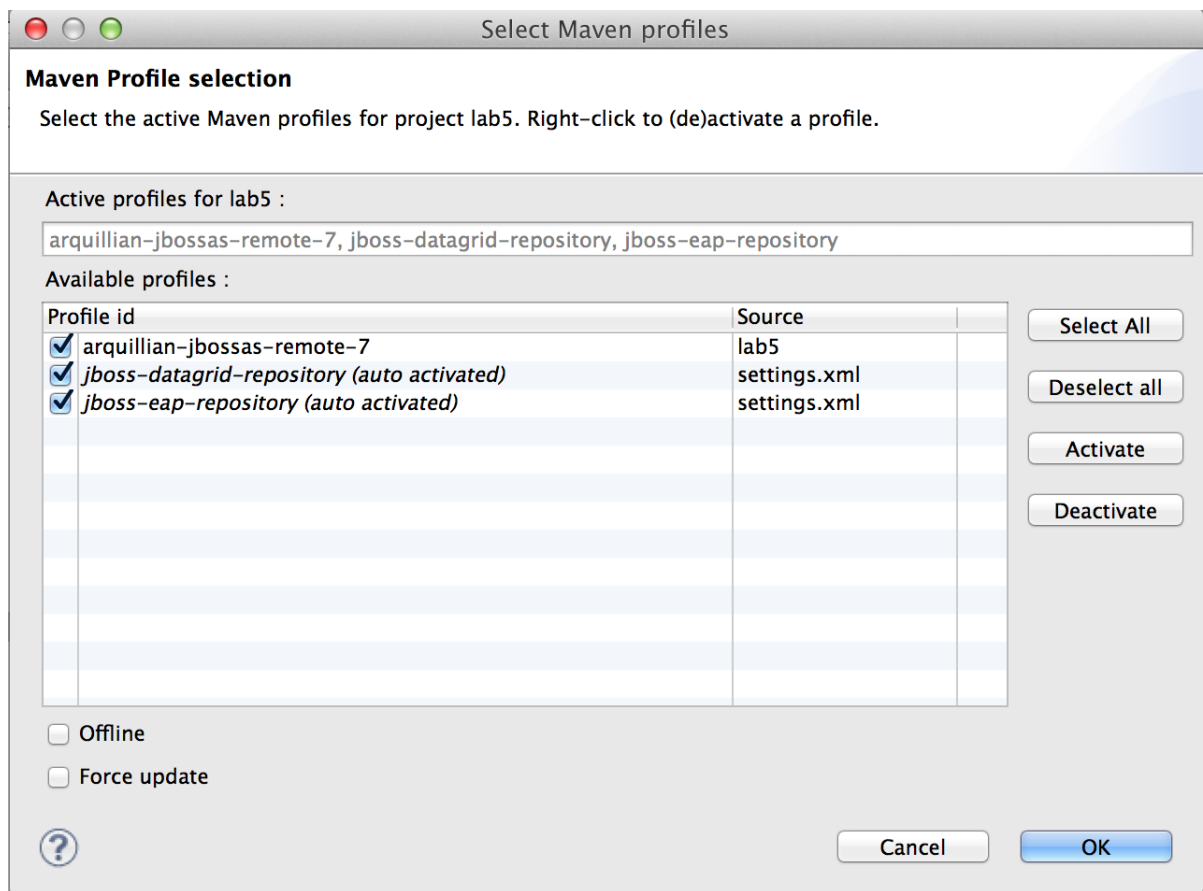
11. You should now have a workbench with warnings, but without problems.

12. You are now ready to start with lab1