Lab 3 - Building with Maven Tycho

The goal of this lab is to set up a build process for the application based on Maven Tycho.

Start by pointing Eclipse at the `webapps/root/labs/lab-3` folder contained in the tutorial root. Import the projects into the workspace.

Make sure that there are no spaces in the path to this workspace. A bug in the Maven Tycho publishing process will cause problems if you have spaces in your path.

In this lab you will do the following:

1. Build the core application using Tycho.

2. Build the additional feature using Tycho.

Build the core application using Tycho.

Prepare Maven and repositories for the lab.

Verify that Maven is correctly installed by entering `mvn -version` at a command prompt. Check that you get a valid response and that the version of Maven is 3.0 or greater.

If Maven is not correctly installed, you may need to install Maven 3 or configure your path so that it's on your system path. An archive containing Maven 3 can be found in the `webapps/root/maven-dist` folder. You can also download it directly from the index page of this Tutorial's web server.

Create a release engineering project and build the bootstrapper feature.

Note that we have modified the `eclipse.product` file to enable it to work with Maven and p2 repositories. Start-level and auto-start information has been added on the product **Configuration** page. Without this Maven Tycho will not create a proper `config.ini` file.

1. Select **File > New > Project…** from the main menu. Select **General > Project** from the list of wizards and click **Next**.

2. On the first page of the wizard, enter `com.example.app.bootstrapper.releng` as the **Project name**. Leave everything else as is and click **Finish**.

3. Copy the file `pom.xml` from the `extra-files` project into the new project.

4. Open this new `pom.xml` file. Locate the `configuration` element for the `target-platform-configuration` plugin. Comment out or delete the environments that are not appropriate for your machine. Note that you may need to add `_64` to the end of the `arch` element if you're running a 64 bit JVM. This has been done for the Mac OS X environment already.

5. From a command prompt, switch to the `com.example.app.bootstrapper.releng` folder in your workspace. Enter `mvn clean package` and verify that the build completes successfully.

   Note, though that the build doesn't actually do anything yet because the parent pom is not yet linked to it's child projects.

6. The other bootstrapper projects already have pom.xml files. Examine these, paying particular attention to the `packaging` elements.

7. Link the child projects to the releng project by adding these `module` elements in the empty `modules` element.

```
<module>../com.example.app.bootstrapper.feature</module>
<module>../com.example.app.bootstrapper</module>
<module>../com.example.app.bootstrapper.test</module>
```

8. Run `mvn clean package` from the command line and the build should complete successfully.

Add the product to the build.

1. Open the `eclipse.product` file and enter `com.example.app.bootstrapper.product.id` into the **ID** field under **General Information**. This field is necessary for the Maven p2 director plugin to function properly.

2. Open the parent `pom.xml` file in the `com.example.app.bootstrapper.releng` project. Locate the `modules` element and add a new `module` sub-element that references the `com.example.app.bootstrapper.product` project.

```
<module>../com.example.app.bootstrapper.product</module>
```

3. Re-run the build from the command line. The build should complete successfully.

Install and run bootstrapper.

1. Refresh the `com.example.app.bootstrapper.product` project and locate the `target/products` folder. In this folder is a zip archive containing the application.

2. Extract this archive somewhere on your local machine.

3. For Mac OS X users only, Maven Tycho does not create a properly configured `eclipse.ini` file. Locate the appropriate INI file in the `extra-files` project and copy it into the extracted application. Rename it to `eclipse.ini`.

   In Finder, you will need to right click on the `Eclipse` application bundle and select **Show Package Contents** on the context menu. The INI file is located in the `Contents/MacOS` directory.

   For those of you interested in why this is necessary, see the following bug report:

   https://issues.sonatype.org/browse/TYCHO-595

4. Launch the application and note that you get a warning dialog. The perspective cannot be found because the update site has not been built yet. We'll do that next. Shut down the launched application.

Build the additional feature using Tycho.

A `com.example.app.perspectives.releng` project has been created for you and it contains a parent `pom.xml` file. The other projects related to this feature have a `pom.xml` file as well. Take a few minutes to examine the pom files.

1. Before doing a build, we need to repair the `site.xml` file in the `com.example.app.perspectives.p2` project. This file is modified when doing a **PDE Build > Build Site** command, and the change causes the Maven Tycho build process to fail.

   To fix the problem, remove `com.example.app.perspectives.feature` from the list in the `site.xml` file and re-add it.

2. In a terminal window, navigate to the `com.example.app.perspectives.releng` directory. Do a `mvn clean package` and the build should complete successfully.

3. Refresh the `com.example.app.perspectives.p2` project. You'll see the update site has been created as both a folder and as an archive.

4. Re-run the application. The new feature should be installed from the update site and the perspective should now appear. Delete the installed application. This lab is now complete.

   Note that the URL system argument has been modified to point to the new update site location. Open `eclipse.product` and examine the arguments on the **Launching** tab to see this.