**Setting Up the Development Environment for the Heatseq Tool in Eclipse**

**Section 1: Eclipse Setup**

1. Download the current **Eclipse IDE for Java Developers** (Juno) from <http://www.eclipse.org/downloads/>.

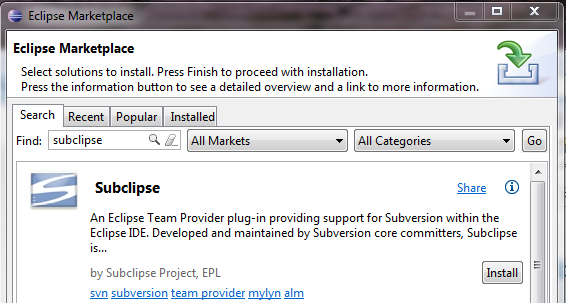
2. Install Eclipse by unzipping the downloaded file into the desired install location (Recommended: C://Eclipse/Juno/). Note: If you install in Program Files you may have to run as Admin to get plugins to install correctly.

2A (optional) – Deleting the C://Users/[user name]/.eclipse directory may alleviate issues with installing plug-ins.

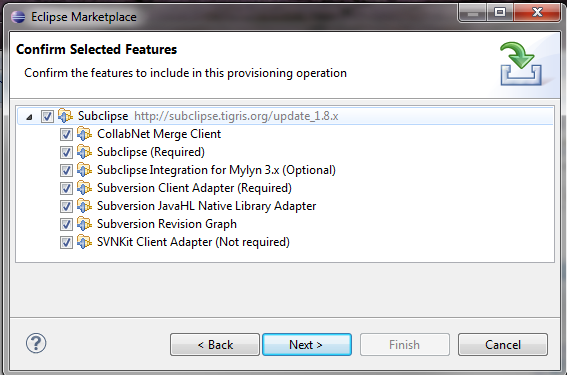
3. Run eclipse by double clicking the eclipse.exe file (sample location: C://Eclipse/Juno/eclipse/eclipse.exe). You may want to right click the eclipse.exe file and “Run as Administrator” if you run into issues installing plug-ins.

4. Create a workspace by browsing to the desired location for the heatseq code when the “Select a workspace” dialog appears (Recommended: C://Eclipse\_Workspaces/heatseq\_workspace).

5. Install subclipse plug-in by clicking “Help”>>”Eclipse Marketplace…” from the main menu within Eclipse. Type “subclipse” in the find textbox and click on the “Go” button.



The first search result should be the Subclipse plug-in. Click the “Install” button for the subclipse plug-in.



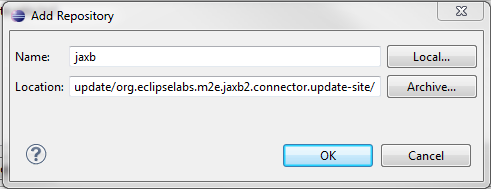
When the “Confirm Selected Features” dialog appears, click “Next”. Agree to the license agreement and click “Finish”. Restart eclipse when prompted to do so.

6. Install m2e-jaxb2-connector by clicking on Help>Install New Software. Type the following into the “Work with” drop down box:

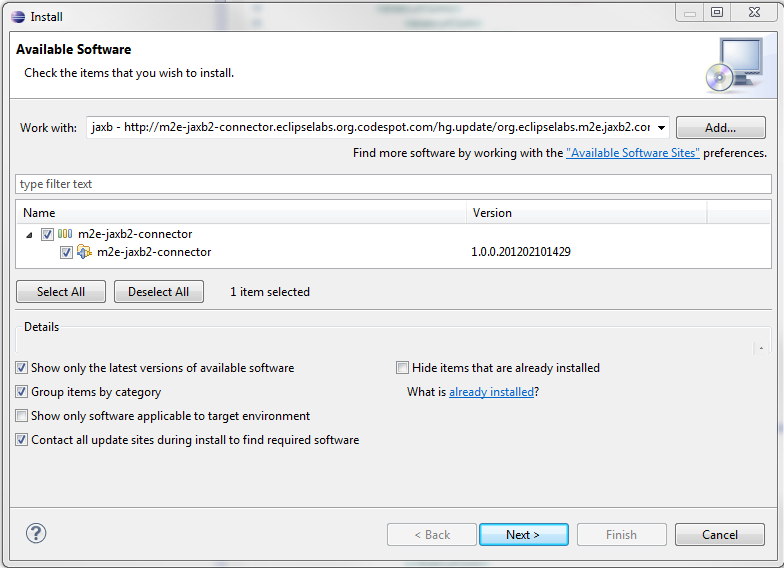
http://m2e-jaxb2-connector.eclipselabs.org.codespot.com/hg.update/org.eclipselabs.m2e.jaxb2.connector.update-site/

Click the “Add…” button:

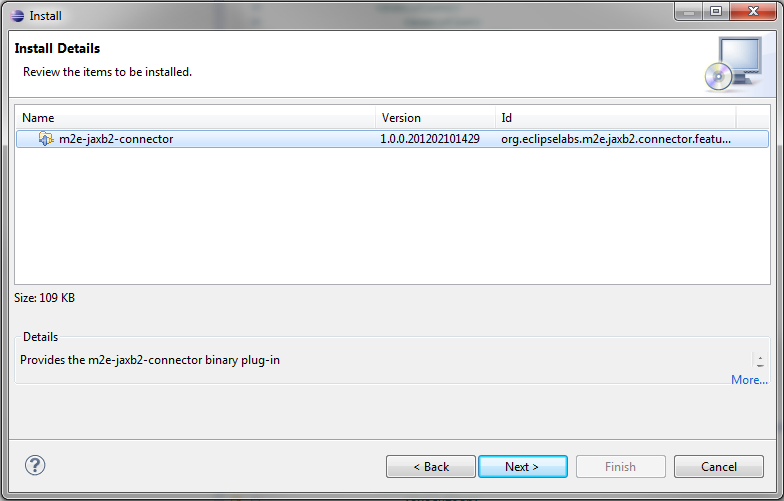
Type an arbitrary name in the “Name” textbox when the “Add Repository” dialog appears and click the “OK” button.



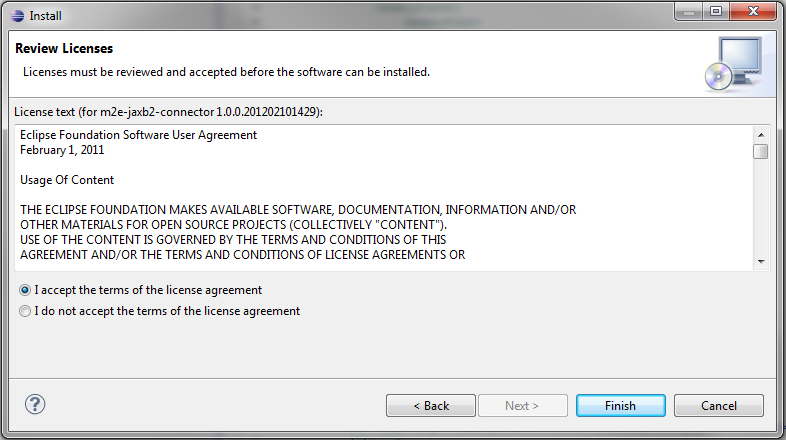
Make sure the m2e-jaxb2-connector is selected and click the “Next>” button.



An “Install Details” panel will appear, click the “Next>” button.

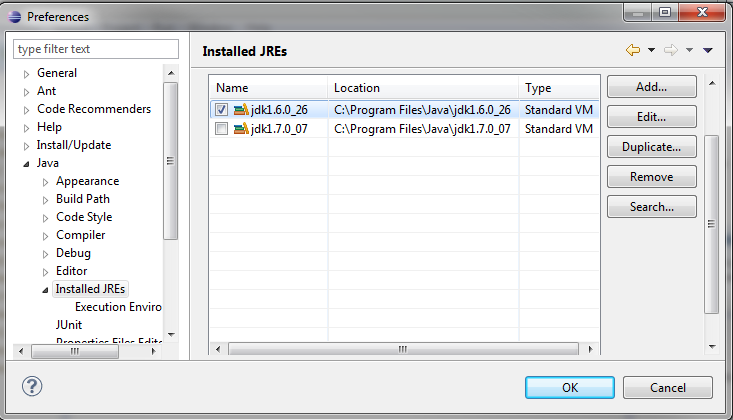


Make sure the “I accept the terms of the license agreement” radio button is selected and click the “Finish” button.



After the connector is installed you will be prompted to restart eclipse, comply.

7. Set a java 1.7 jdk in Eclipse to the default by clicking “Window”>>”Preferences” from the main menu. Open the “Java”>>”Installed JREs” panel. If there is not a jdk1.7 listed, add one by clicking the “Add...” button. Select “Standard VM” and click “Next”. Click on the “Directory…” button and browse to an installed 1.7 jdk (For example: C:\Program Files\Java\jdk1.7.0\_26) and click the “OK” button in the “Browse For Folder” dialog. Click the “Finish” button in the “Add JRE” dialog. Make sure that the 1.7 jdk has its checkbox selected and click “OK”.



8. Optional—If you would like to dictate where all the Jar dependencies will be downloaded continue, otherwise the default location will be used .

Set the maven repository location by clicking “Window”>>”Preferences” from the main menu. Open the “Maven”>>”User Settings” panel. Select a maven repository location by clicking the “Browse…” button located to the right of the “User Settings” text box. Locate an appropriate settings.xml file. If an appropriate file cannot be found save a file with the content found below as “settings.xml”:

<settings xmlns="http://maven.apache.org/settings/1.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/setrings/1.0.0

<http://maven.apache.org/xsd/settings-1.0.0.xsd>">

<localRepository>c://454\_maven\_repository/.m2/repository</localRepository>

<mirrors>

<mirror>

<id>nexus</id>

<name>Nexus Public Release Mirror</name>

<url>http://ci.dbr.roche.com:8081/nexus/content/repositories/public/</url>

<mirrorOf>\*</mirrorOf>

</mirror>

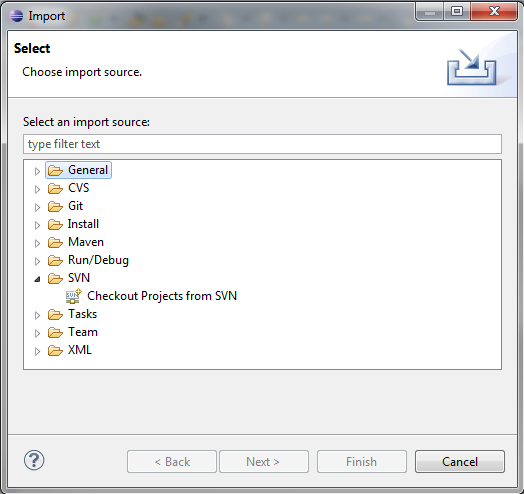
</mirrors>

</settings>

The contents of the “localRepository” tags should be adjusted to point to the desired location for the local repository.

**Section 2: Heatseq Project Setup**

1. Import the Heatseq project into your workspace from subversion by right clicking in the “Package Explorer” and selecting “Import…”. Within the SVN folder, select the “Checkout Projects from SVN” option and click “Next”.



Make sure that the “Create a new repository location” is selected and click the “Next>” button. In the “Url” text box enter <http://svn/repos/sandbox/trunk/heatseq> and click the “Next>” Button. Select all of the folders underneath the repository location (Holding shift while clicking the top and bottom folders will accomplish this goal). Click the “Finish” button (Time Estimate: ~2 minutes).

2. Convert the imported projects to maven projects by selecting them all and selecting “configure”>>”Convert to Maven Project” from the right click menu.

You should now be able to build the Heatseq code base within eclipse via Maven or utilizing eclipse directly (useful for debugging).

**Section 3: Building Heatseq Tool Using Maven within Eclipse**

1. Find the pom.xml files which builds the application via Maven can be found in the “build” project under the “heatseq\_commandline” folder.
2. Right clicking on this pom.xml files and selecting “run as …>Maven Install” will build the heatseq application and all of its modular dependencies. Artifacts produced by the build can be found in the ”nimblgen\_heatseq” project “target” directory (you may need to refresh the projects before artifacts are visible in the target directory by pressing the F5 key with the desired project selected).

**Section 3: Running the Heatseq Tool within Eclipse**

1. Run the heatseq tool within eclipse by right clicking on the “PrefuppCli.java” found in the “com.roche.heatseq.process” package within the “nimblegen\_heatseq” project and selecting “Run As>Java Application”.
2. The first time you run the app it will error out because the application arguments have not been set. You can set these arguments by right clicking on the same “PrefuppCli.java” file and selecting “Run As>Run Configurations…”.
3. Click on the “Arguments” tab and enter the appropriate Program arguments. You will also have to set the VM arguments for memory to ensure that the application has enough memory (for example, -Xmx4g –Xms4g will set a max heap size of 4 gigabytes with an initial heap size of 4 gigabytes).



**Section 4: Setting Up Auto-Formatting and Code Style Sheet for Editing Code**

Setting Up Your Java Formatting Preferences

To ensure that everyone on the team is using the same Java code formatting the Eclipse Java Formatter preferences need to be set.

1. Within Eclipse, select “Window”>”Preferences”.
2. Expand the “Java>Code Style>Formatter” section in the displayed tree.
3. Select the “Import” button and locate the file that will be in your workspace at “\superpom \eclipse\_formatting\_v1\_0.xml”.
4. Click the “Apply” button to ensure there are no errors.
5. Expand the “Java>Editor>Save Actions” section in the displayed tree.
6. Check “Perform the selected actions on save” and “Format source code” with the “Format all links” radio button selected.