RCP Development Environment Setup - For Core Velo/Vanilla Velo RCP

- Prerequisites
- Install Java 7
- Install Eclipse
- Import Your Project's Source Code
  - Using Eclipse (Recommended)
  - From outside Eclipse (Optional - Advanced developers only)
- Build Your Target Platform
- Setup the Runtime Environment for your RCP Application
  - Create a repository.properties file for your Velo application
  - Create a Run Configuration
- Launch Your Velo RCP Application
- Updating Your Target Platform

Prerequisites

1. Get the default repository.properties file from your Velo Server administrator. (see below for how it is used)

Install Java 7

1. The Velo-RCP now works with Java 7. The architecture you choose (64-bit vs. 32-bit) **MUST** match the architecture of Eclipse that you install (see below).
   a. To determine if you have java installed and what version it is, run this from the command line:
      java -version
   b. If you get a "command not found" error - you do not have java installed
   c. If your version says anything except 1.7.*, you need to install Java 7

Install Eclipse

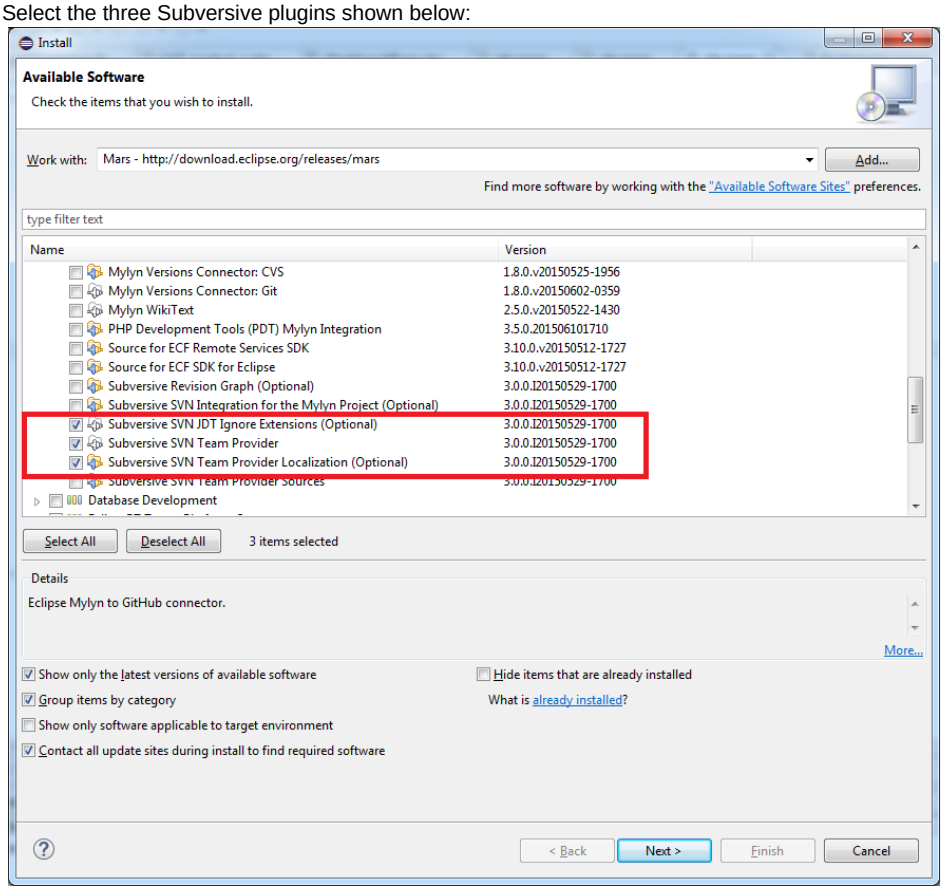
1. Download **Eclipse MARS for RCP developers** (https://www.eclipse.org/downloads/packages/eclipse-rcp-and-rap-developers/marsr) (or for PNNL developers look in \\pnl\projects\CAT_Tools\Eclipse)
   a. Decide if you want to use a 32-bit Eclipse or 64-bit Eclipse (we hope 64 bit). You **MUST** have the same version of Eclipse as the Java you install!
2. Start Eclipse and provide it with a workspace path. The workspace is where all Eclipse projects will be saved. We recommend using a name that matches your Velo deployment, since many Velo developers work on multiple projects. *(For example, my Akuna workspace is saved to C:\Eclipse\Workspaces\Akuna.)*

   > We will hence refer to your workspace folder as $WORKSPACE

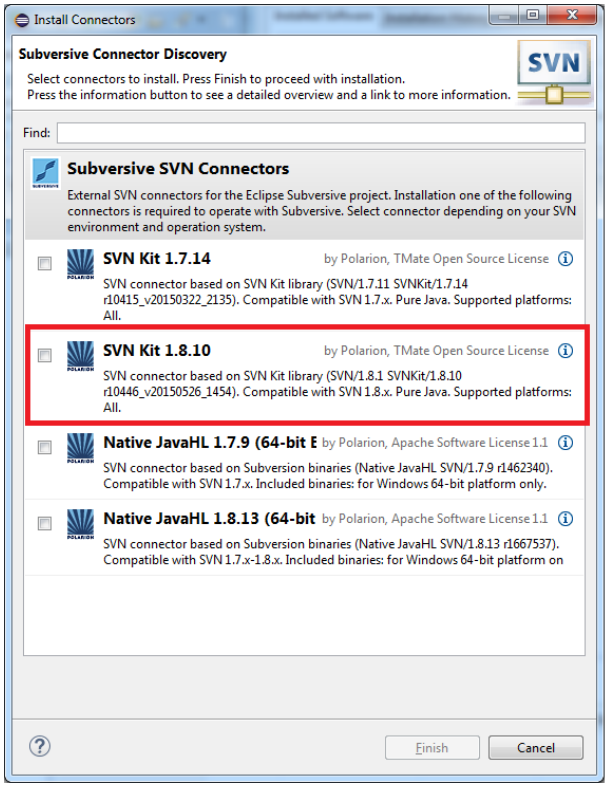
3. Close the welcome page if it is opened.
4. Set Java version to point to the java from Prerequisites #1
   a. Open preferences
      i. (Mac) Eclipse->Preferences
      ii. (Windows) Windows->Preferences
   b. From the left navigation bar, select Java->Installed JREs
   c. If the JDK is not found in the list you may add a new one
      i. Select Add...
      ii. Select Standard VM
      iii. Set JRE home (e.g. C:\Program Files\Java\jdk1.7.0_71)
      iv. Set JRE name
      v. Finish
   d. Select the preferred JDK by checking the box on the left
   e. Click OK
5. (Optional) Configure Proxy Settings. If you have a proxy at your site, you will need to configure those proxy settings in Eclipse so you will be able to access the plugin and svn repositories. Follow these instructions: http://help.eclipse.org/mars/index.jsp?topic=%2Forg.eclipse.

platform.doc.user%2Freference%2Fref-net-preferences.htm

6. (Optional) Install the **subversive plugin.** This is useful so you can check out/commit your code changes directly from Eclipse and view changes in the compare editor. **It is recommended for novice users.**
    a. In Eclipse, Help->Install New Software
    b. In the Work with: box, select "Mars - http://download.eclipse.org/releases/mars" from the dropdown list
    c. Expand the Collaboration box
    d. Select the three Subversive plugins shown below:

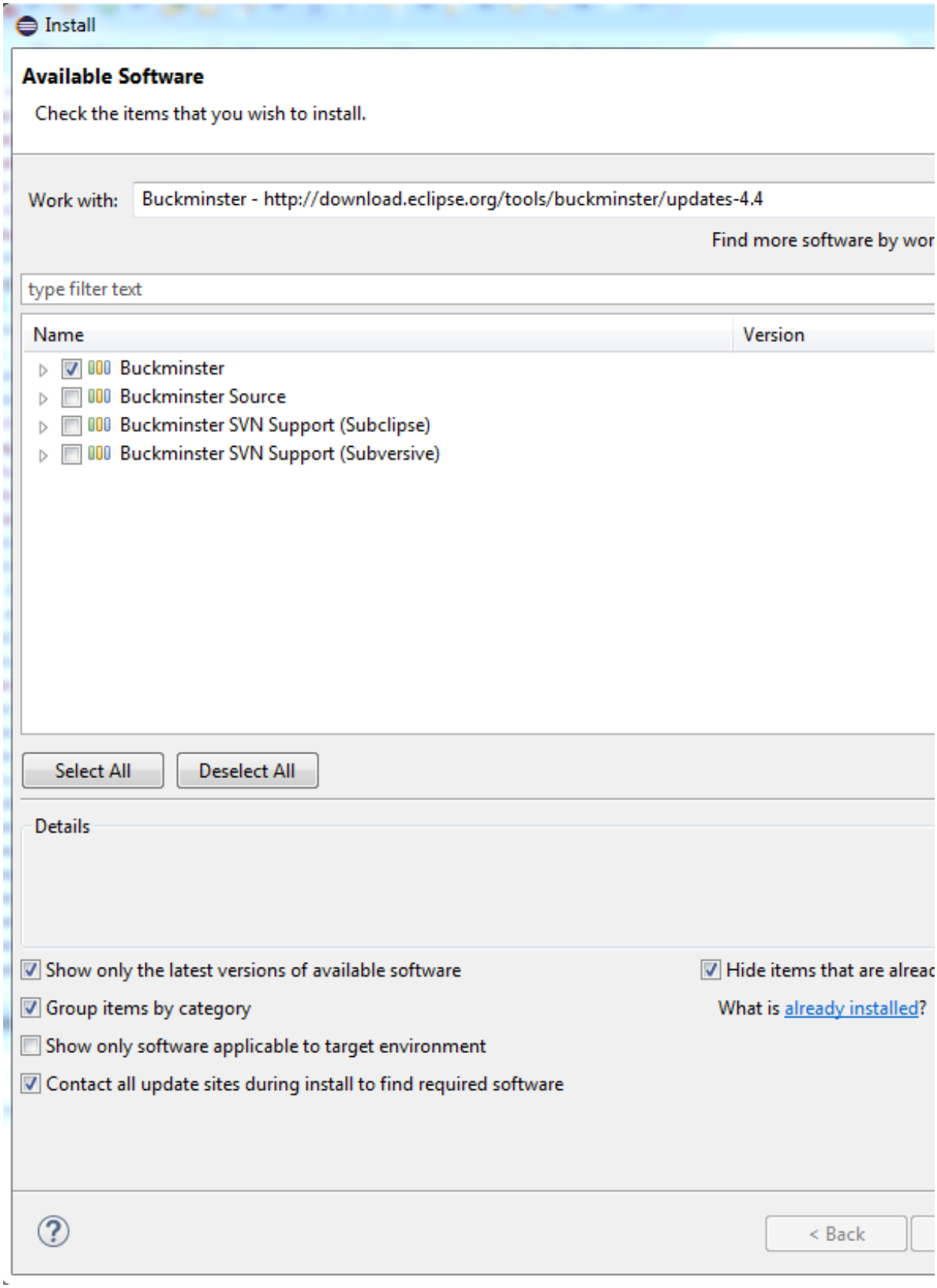


    e. Click Next twice, accept the user agreement, hit Finish, and restart Eclipse when prompted
    f. The first time you try to navigate to the SVN Repository Exploring perspective (Window->Open Perspective->Other, See below when checking out source code), you will be automatically prompted to select an SVN Kit Connector. Select 1.8.10 as shown:



g. Click Next twice, accept the user agreement, hit Finish, and restart Eclipse when prompted.

7. Install Buckminster (This step will soon be phased out once all builds are done using Tycho instead)
   a. From the Eclipse main menu, select Help->Install New Software
   b. In the Work with: box, type http://download.eclipse.org/tools/buckminster/updates-4.4
   c. Click the Add button, which pops up another window so you can type the name. In the name box, type **Buckminster 4.4**
   d. Check Buckminster
   e. The form will look like the example below:



f. Click Next twice, accept the user agreement, hit Finish, and restart Eclipse when prompted

8. Add Tycho configurator to Maven. Go to Preferences->Maven->Discovery. Click on the Open Catalog button. Type Tycho in the filter

box. Check Tycho and follow the steps to install, which will require a restart of Eclipse.

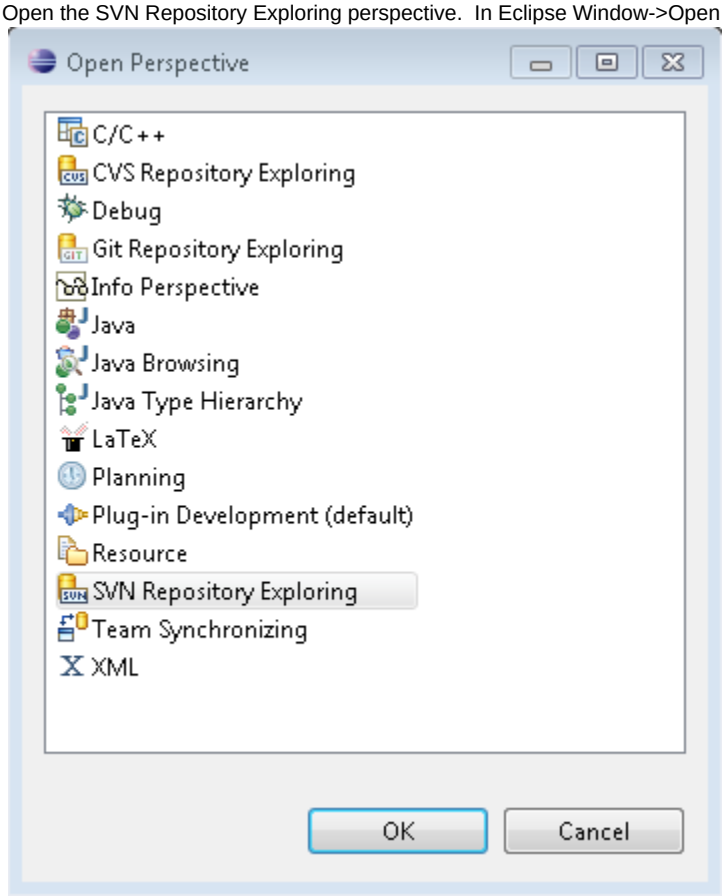
Import Your Project's Source Code

You can import source code from either Eclipse or outside Eclipse using a different svn tool. If you are new to Java development, we recommend that you import your code from Eclipse.

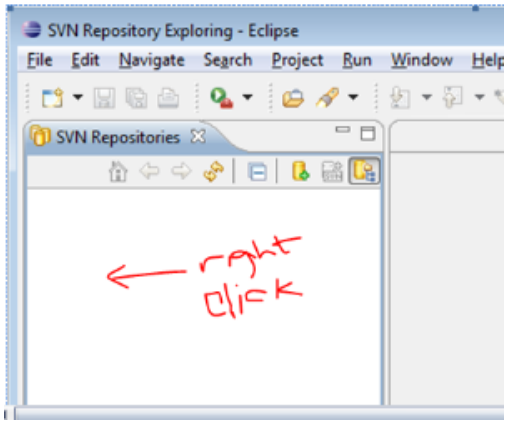
Using Eclipse (Recommended)

Connect to the source code repository in eclipse.

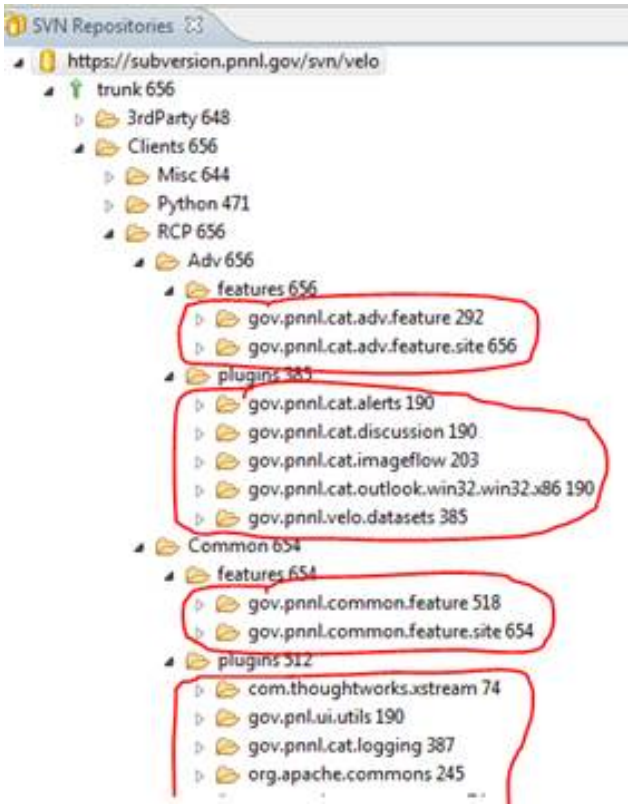
1. Open the SVN Repository Exploring perspective. In Eclipse Window->Open Perspective->Other



2. Right Click in the SVN Repositories view in the upper left corner and select New->Repository Location.



3. Enter the repository URL <TODO Put Public Velo SVN mirror here> in the form and click Finish.
4. You will be prompted for the authentication for your SVN repository.
5. If you get an error regarding "Failed to load JavaHL Library" click "OK" and do the following:
    a. Open preferences from Eclipse->Preferences on Mac or Windows->Preferences on windows
    b. Open Team->SVN
    c. From the "SVN interface" drop down list, choose the option similar to "SVNKit (Pure Java) SVNKit v1.3.5.7406"
    d. Click OK
6. Select the projects to check out (they look like folders) from the SVN Repositories view.
    a. In the SVN Repositories view, expand the folders until you see the projects you are supposed to check out
    b. Hold the control/command key down so you can select multiple projects (a.k.a. folders)
    c. After you have selected the correct projects, right click and select the Checkout context menu item
    d. Click the Finish button to checkout the projects to your workspace (all default settings are ok)
    e. Check out these projects <TODO CHANGE THESE SCREENSHOTS ONCE PUBLIC SVN MIRROR IS CREATED>:



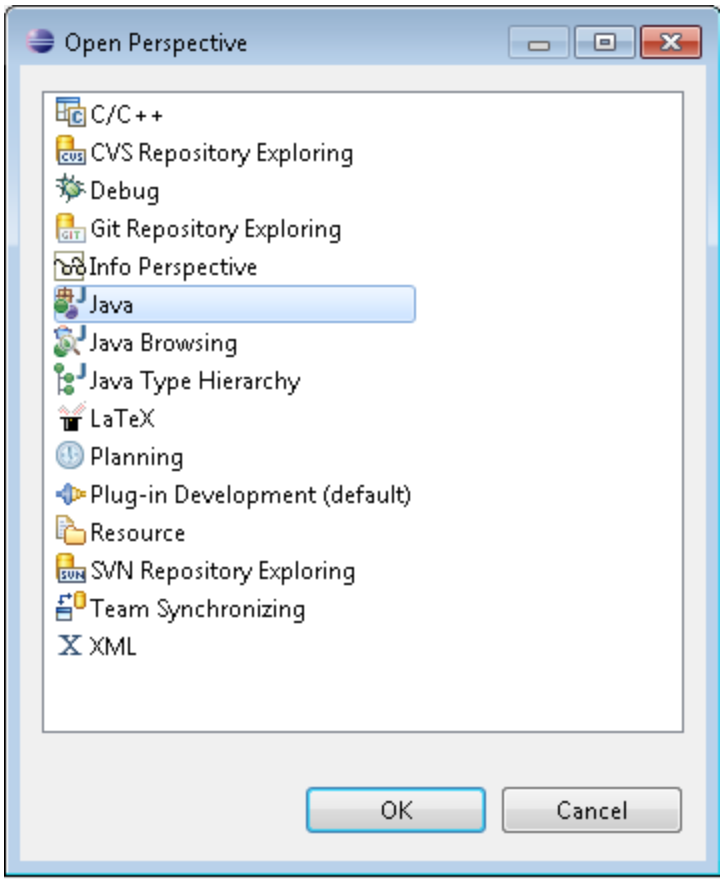
From outside Eclipse (Optional - Advanced developers only)

1. Checkout the source using your preferred tool (e.g. Tortoise SVN)
2. Open the Java Perspective. In Eclipse, Window->Open Perspective->Other
3. In Eclipse, File -> Import
4. Choose General -> Existing Projects into Workspace and hit Next
5. Browse to the folder where the source files are checked out.
6. Select the projects from step 6e above
7. Click Finish.
8. Projects are now imported into Eclipse and should appear in the Package Explorer view.

Build Your Target Platform

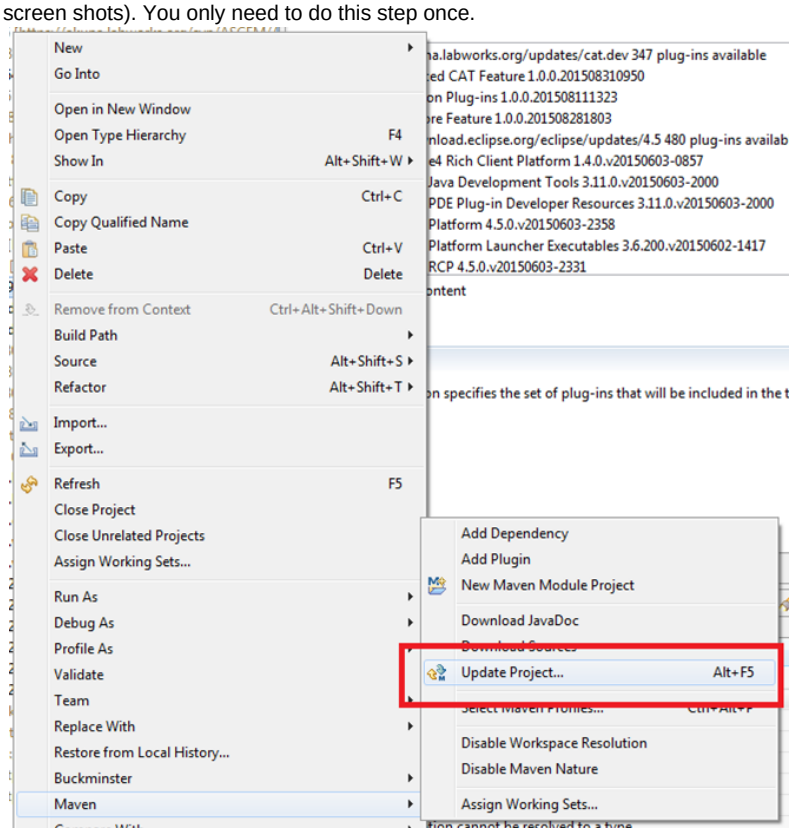
In Eclipse, the target platform contains the set of plugin library dependencies that your application needs in order to compile. The Velo target platform plugins are automatically downloaded from the Velo update site currently located at http://p2.velo.pnnl.gov/updates

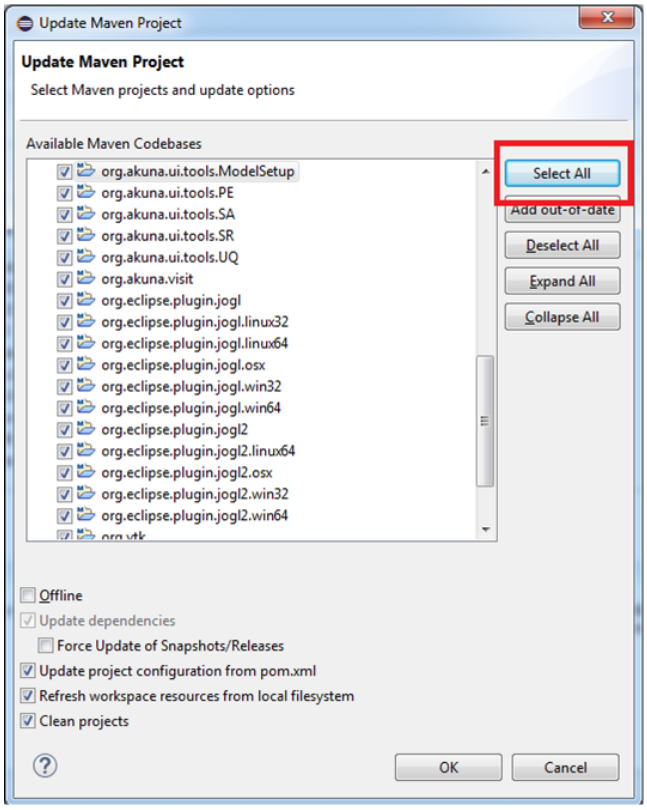
1. Open the Java perspective. In Eclipse Window->Open Perspective->Other



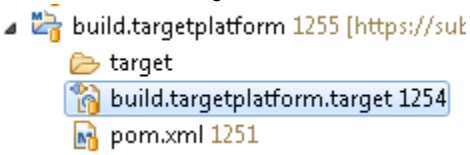
2. Update metadata cache. Right click on any velo maven plugin (you will see the M decorator in the upper left corner), then select Maven->Update Project… Then in the dialog that pops up, click the Select All button, then click the OK button (shown in the following

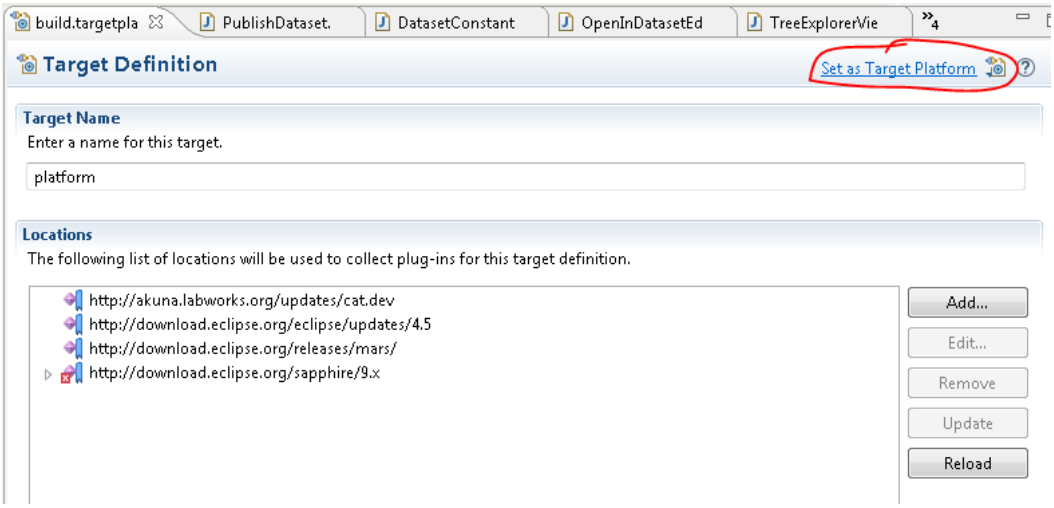
screen shots). You only need to do this step once.





3. Double click on the build.targetplatform.target file in the build.targetplatform project. This should open a special editor as shown below. Click on the "Set as Target Platform" link. Close the build.targetplatform.target file.





4. If there are compile no errors, you are ready to begin development!

Setup the Runtime Environment for your RCP Application

> **YOU ONLY NEED TO DO THIS THE FIRST TIME.**

Create a repository.properties file for your Velo application

The repository.properties file tells the Velo core plugins how to connect to your server. You MUST have a repository.properties file in order to start your Velo application.

> Get the repository.properties file from your project administrator.

2. Create a new file called repository.properties.APPLICATION_NAME using any text editor, where APPLICATION_NAME is the name of your Velo application (e.g., Akuna, PremierNetwork, etc.). An example looks like the following (variables will change per deployment):

```
#
# Server properties
#
hostname=akuna.labworks.org
http.port=80
notification.server.id=akunaDevAlfresco

#
# Generated properties
#
http.url=http://${hostname}:${http.port}
http.url.alfresco = ${http.url}/akunaDevAlfresco
jms.url=http://${hostname}/activemq2

# Required by AlfrescoWebserviceFactory
webservice.location=${http.url.alfresco}/api

wiki.url=http://akuna.labworks.org/akuna-dev/wiki/index.php
```

3. Save the file and close it, Remember the absolute path to this file.

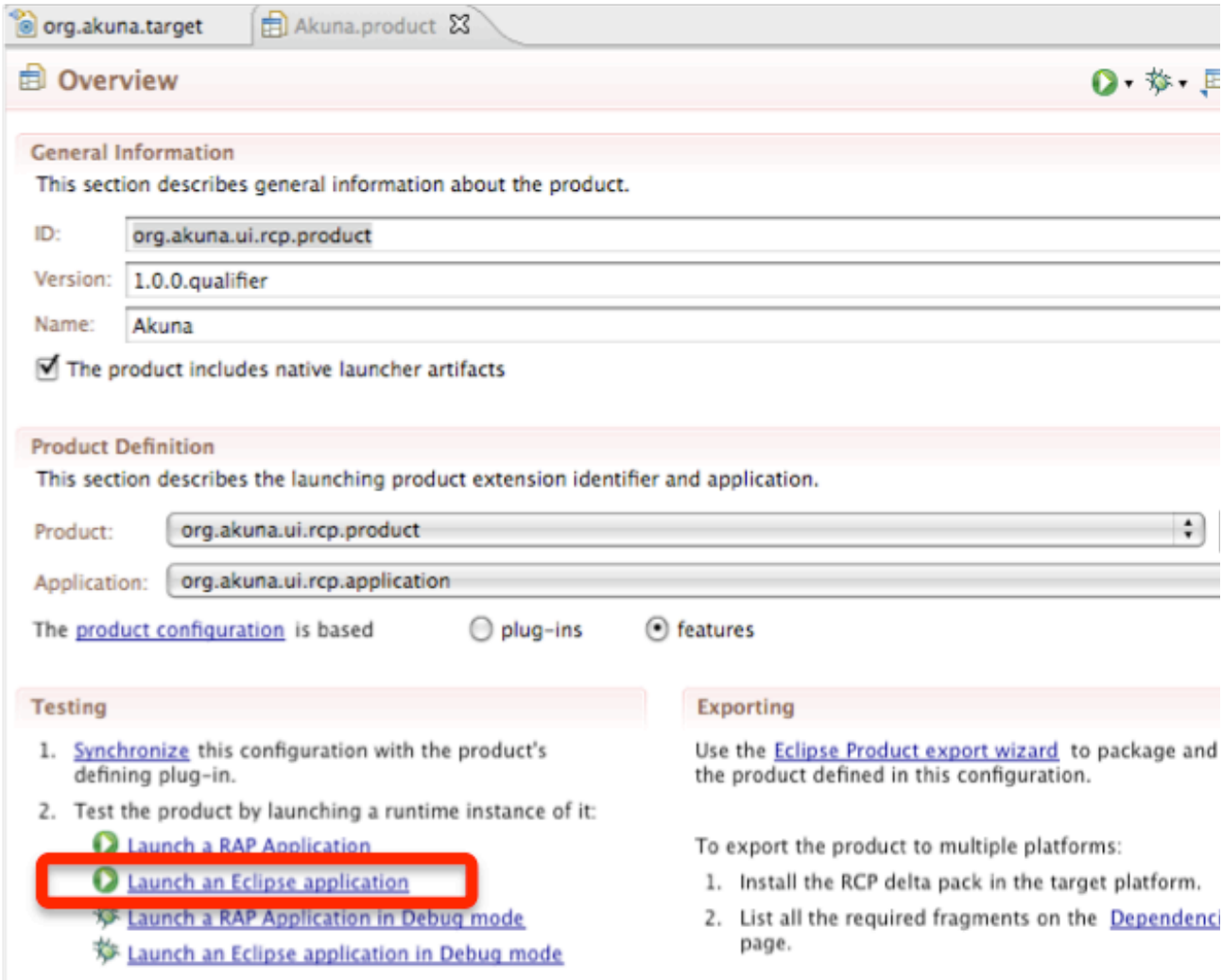
Create a Run Configuration

You need to create a Run Configuration in Eclipse so that Eclipse knows which product to run, which plugins to include, and the command-line arguments to set.

1. In the Package Explorer view in Eclipse, expand the *.rcp project (for example org.akuna.ui.rcp).

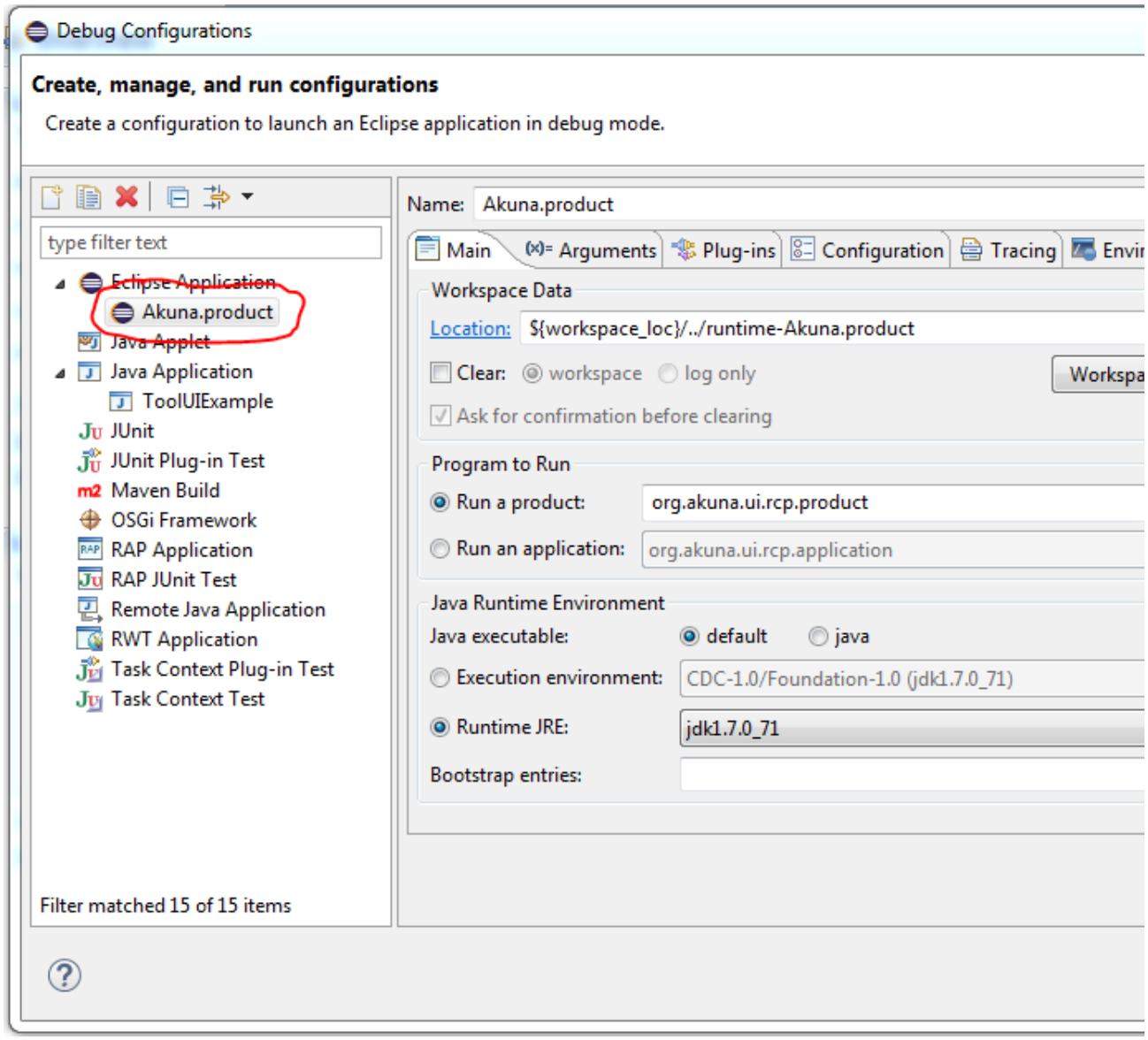
> **DO NOT CHANGE ANY FILES IN THE *.rcp project**! If you do so, you will invalidate the runtime configuration, which will prevent your application from starting up correctly!

2. Open the *.product file (for example Velo.product) - Double click on the file to open
3. Click Launch an Eclipse application

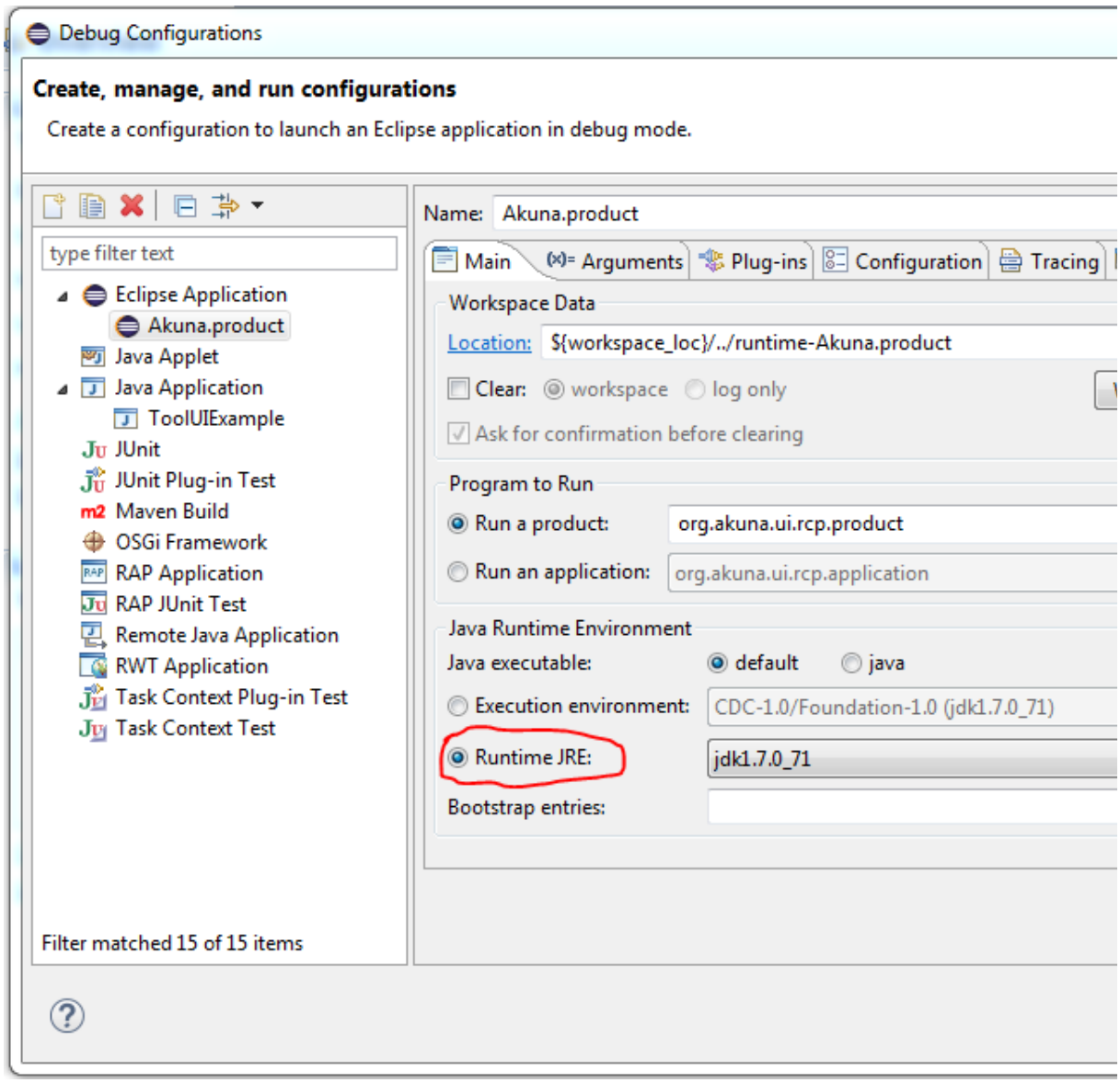


4. It will throw an error, which can be ignored. Click No when prompted to view the log file.
5. From the Eclipse main menu, select Run -> Run Configurations

6. In the Run Configurations window, under Eclipse Application select the *.product for the project you are working on. An example is shown below



7. Edit the Main tab
    a. Make sure you are running in Java 7. Select the Runtime JRE radio as shown and make sure the dropdown has java 7 selected:

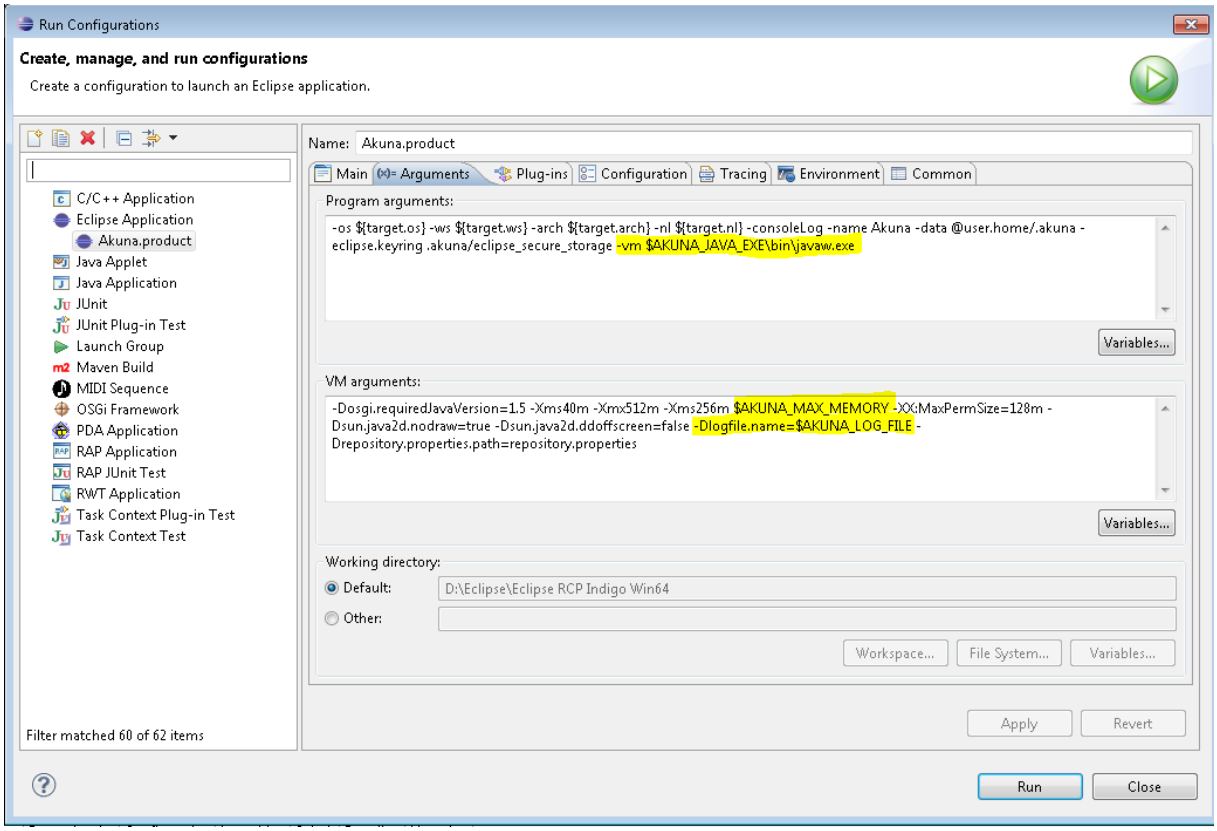


8. Edit the arguments
    a. Select the Arguments tab

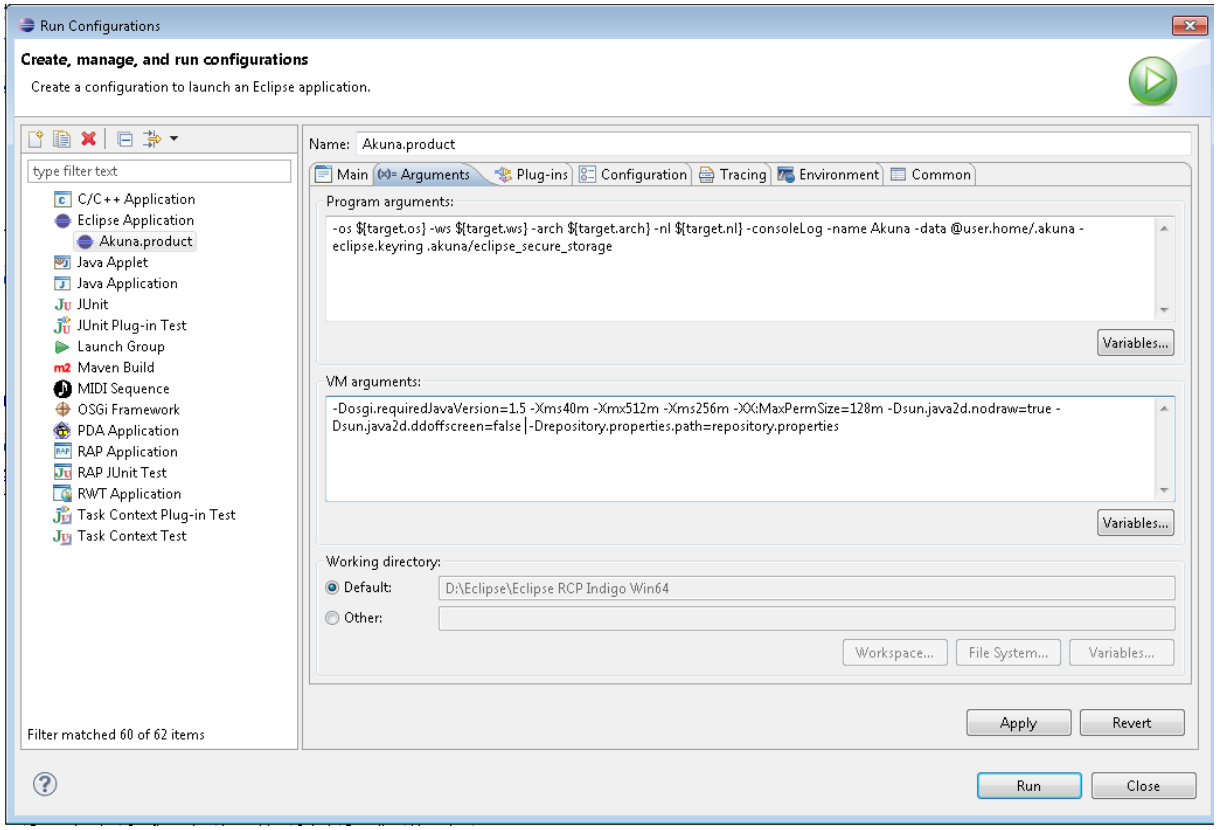
> Confirm the exact arguments specification from your Velo Server administrator if you are running a customized Velo RCP.

b. Remove Variables. Due to variables used by the Velo installer, you should see some weird variables like below (specific variables vary per deployment, but they consist of ALL CAPS and $). In general, **it is safe to delete all variables** from the arguments

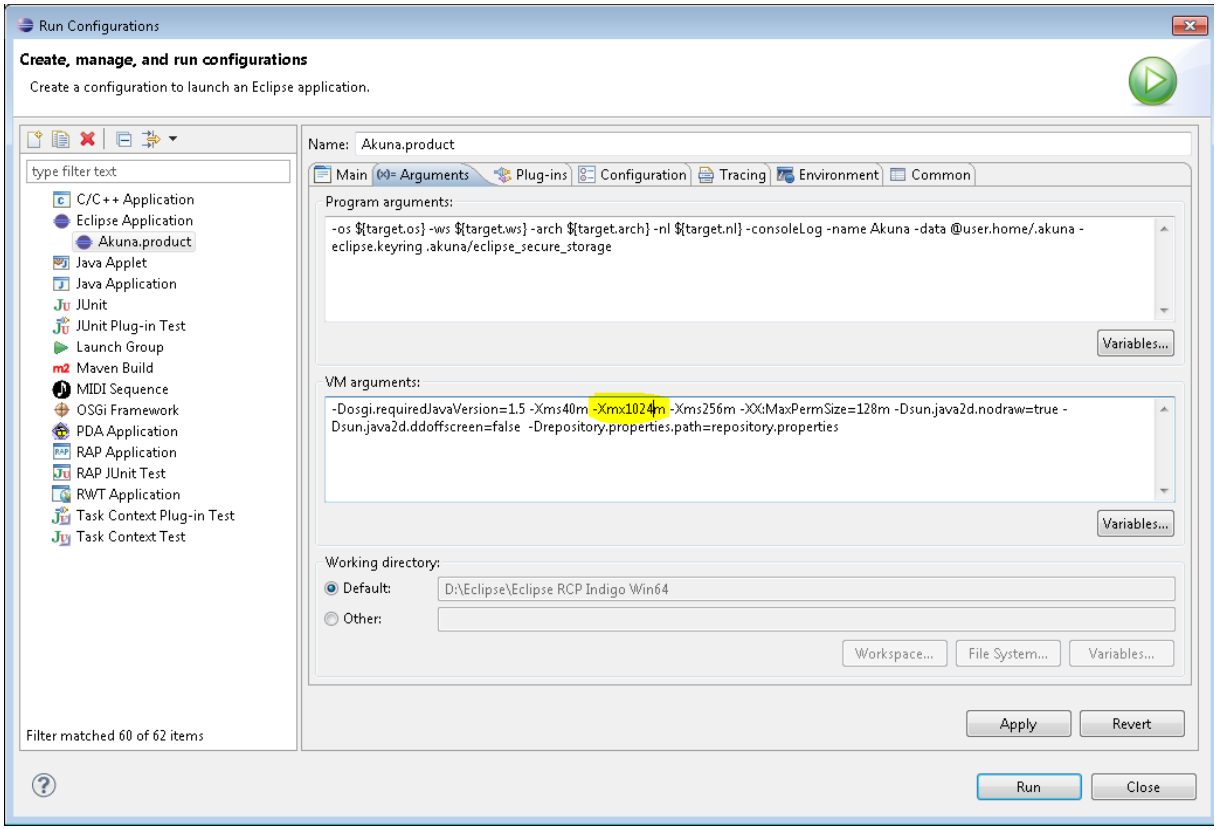
BEFORE



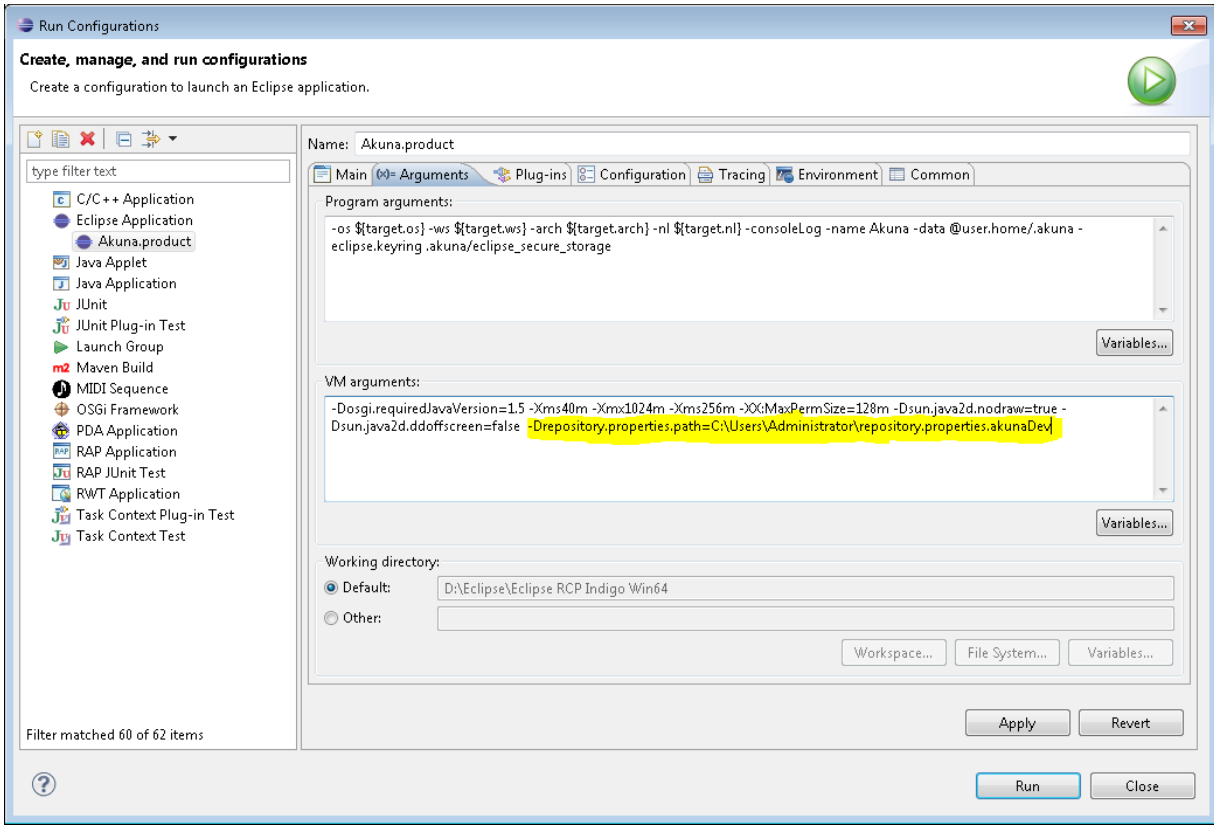
AFTER



c. Set the max memory. We recommend a minimum of 1024 MB

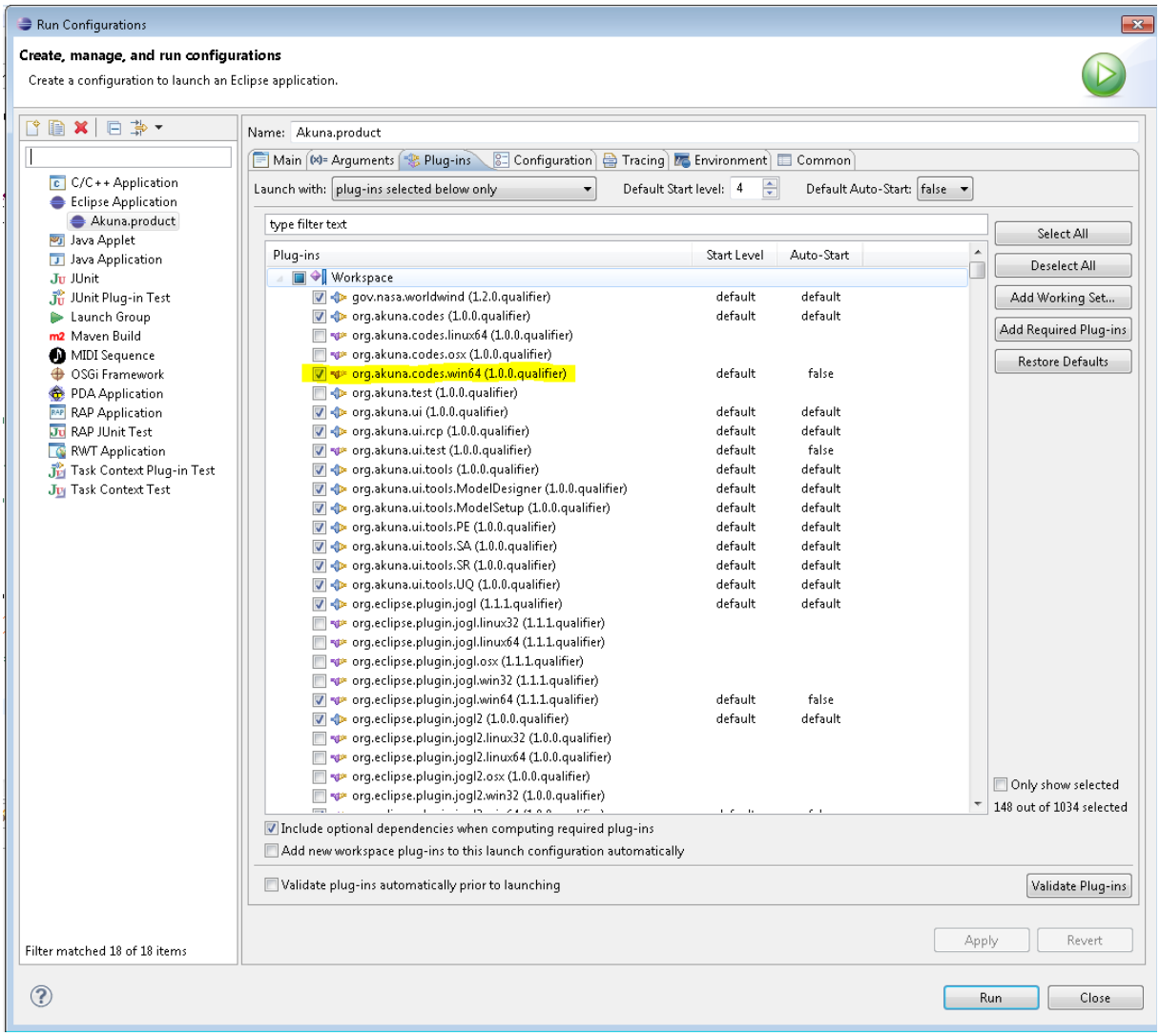


d. Set the path to the repository.properties file to point to where you saved it on your computer.



9. Edit the plugins

    a. Select the Plugin-ins tab (you may need to resize the window in order to see everything in the plugins tab)
    b. Check the Workspace box so that all plugins in your workspace are seleced.
    c. Carefully de-select the plugins that don't apply to your operating system. For example, the akuna.codes plugin has versions for every OS. I am only selecting the one for 64-bit Windows since that is my platform.



d. Click the Add Required Plugins button. This will check for required dependencies under the Target Platform.
e. Validate the selection by clicking on the Validate Plugin-ins button. If you get errors, contact your project administrator for help.

10. Click Apply, then Close

Launch Your Velo RCP Application

1. In Eclipse, click the green Run button. This should automatically start the last Runtime Configuration you had used.

2. If per chance it tries to start a different appliction, then click the drop-down arrow on the Run button and choose the run configuration that was created in the previous section (for example, Akuna.product).

Updating Your Target Platform

From time to time, your project administrator may inform you that you need to update your target platform.

This is because some of the code in the core Velo plugins has changed. If you update from SVN and see strange errors, this is also a good indication that you need to update your target platform. To update your target platform, perform the following steps.

1. Close Eclipse
2. Delete your $WORKSPACE/.buckminster/tp folder
3. Restart Eclipse
4. In the Package Explorer view, expand the *.feature.site project. (For example, org.akuna.feature.site)
5. Double click on site.cquery file this should show the file content on the below. Click on the "Resolve and Materialize" button. Close the site.cquery file.

---

p2.velo.pnnl.gov