DiPro and Prism 3.3.1 Installation from Source Code

In this guide, the installation of DiPro with Prism 3.3.1 version will be shown.Keep in mind that, we use Eclipse Oxygen with JDK version 1.8 and JRE version Java-8-oracle.

Contents

1 Overview

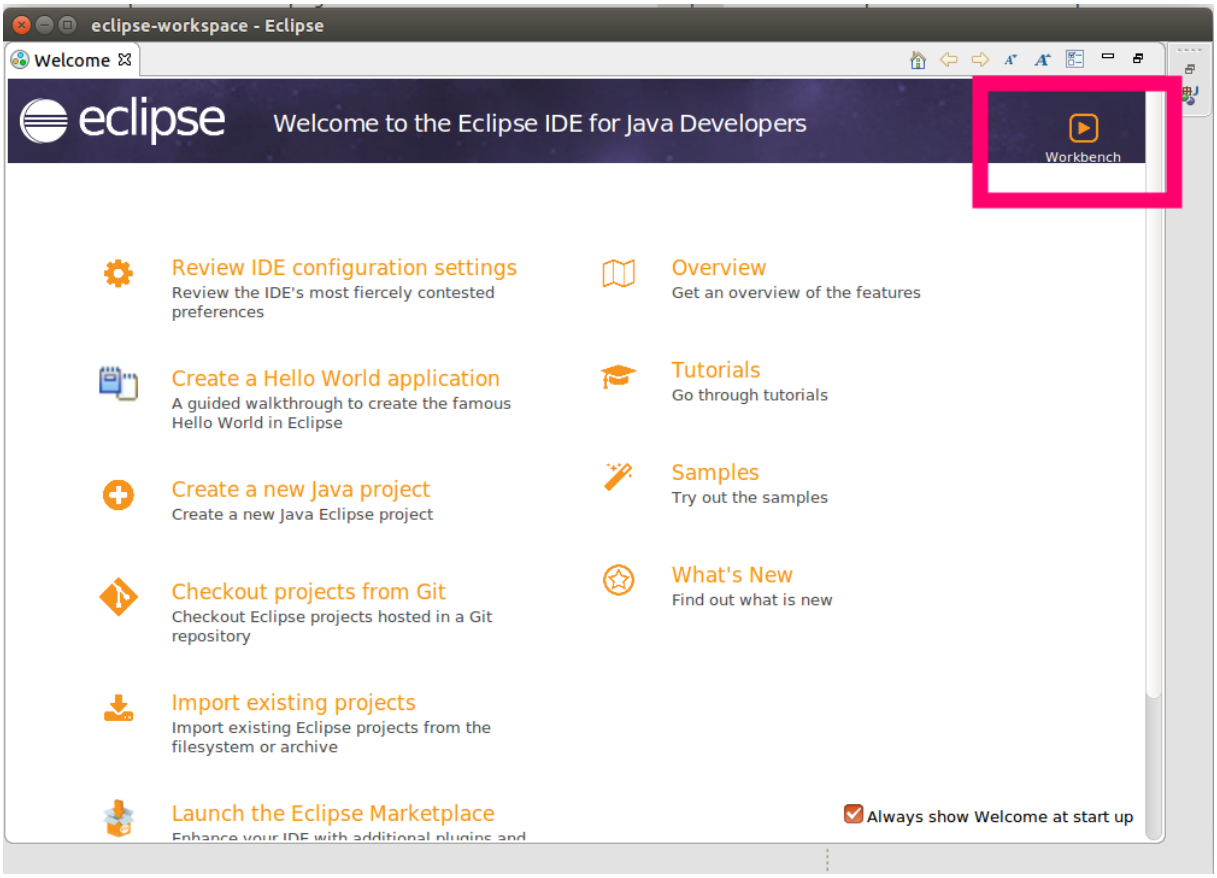
In order to install DiPro to our system we'll need to download **DiPro** source code from svn server of the University of Konstanz. This guide will show step by step installation of DiPro with Prism 3.3.1 in the following order.

- Installing **Subversion** to be able to download svn files from University of Konstanz's website using Eclipse

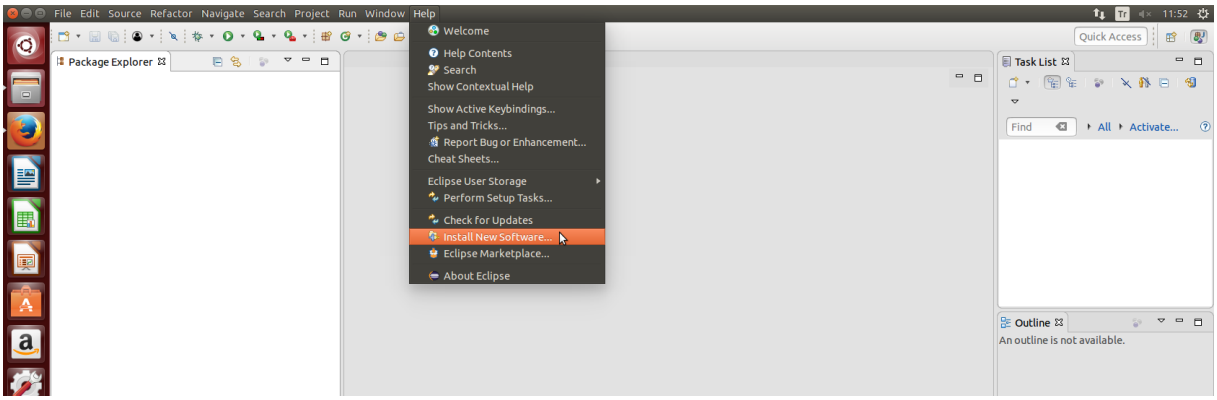
- Downloading the svn files from svn server to Eclipse
- Modifying source folders
- Jre set up
- How to run DiPro
- Frequently Asked Questions

2 Installation of Subversion to Eclipse

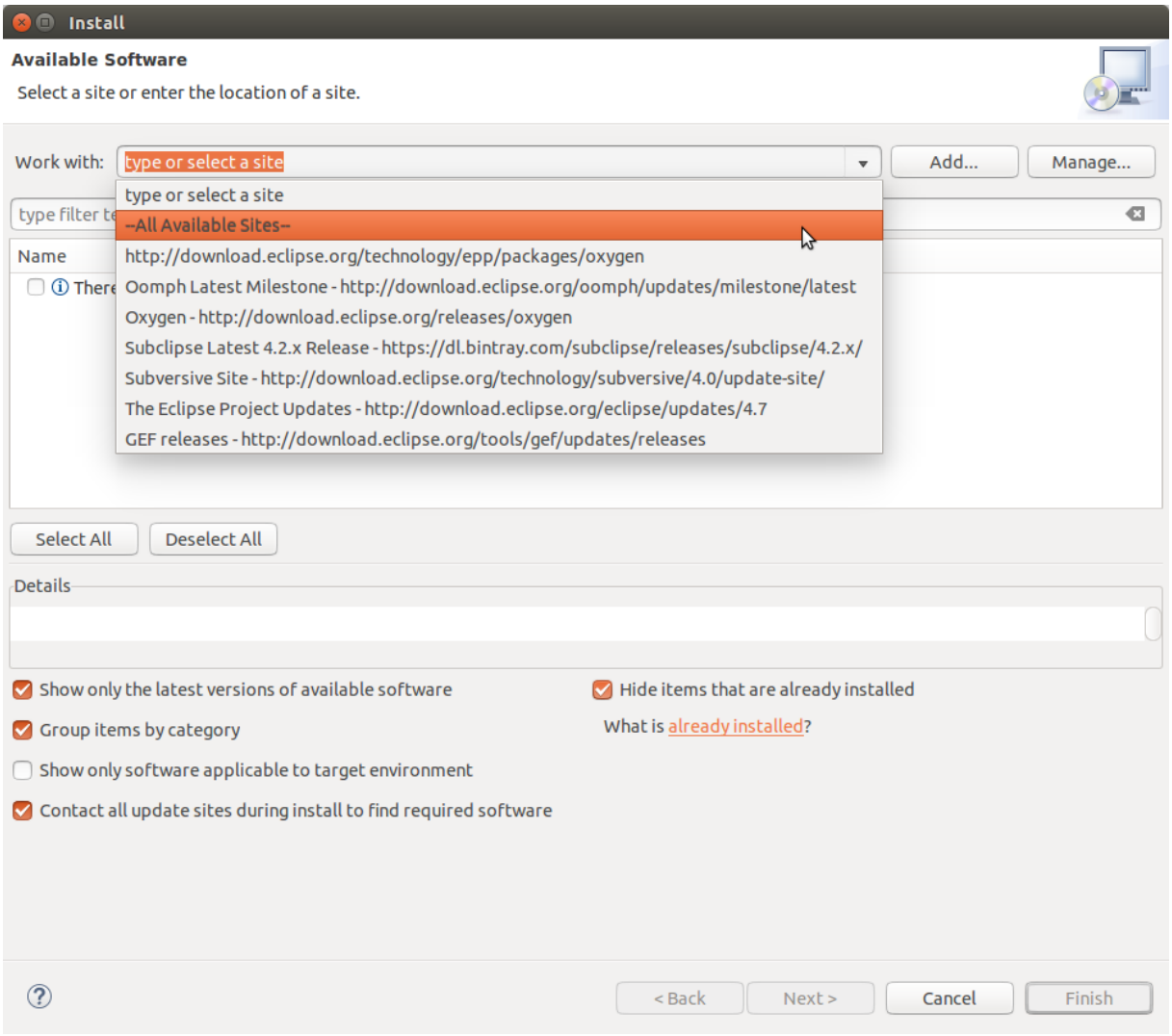
- Open **Eclipse** and change your view to workbench.



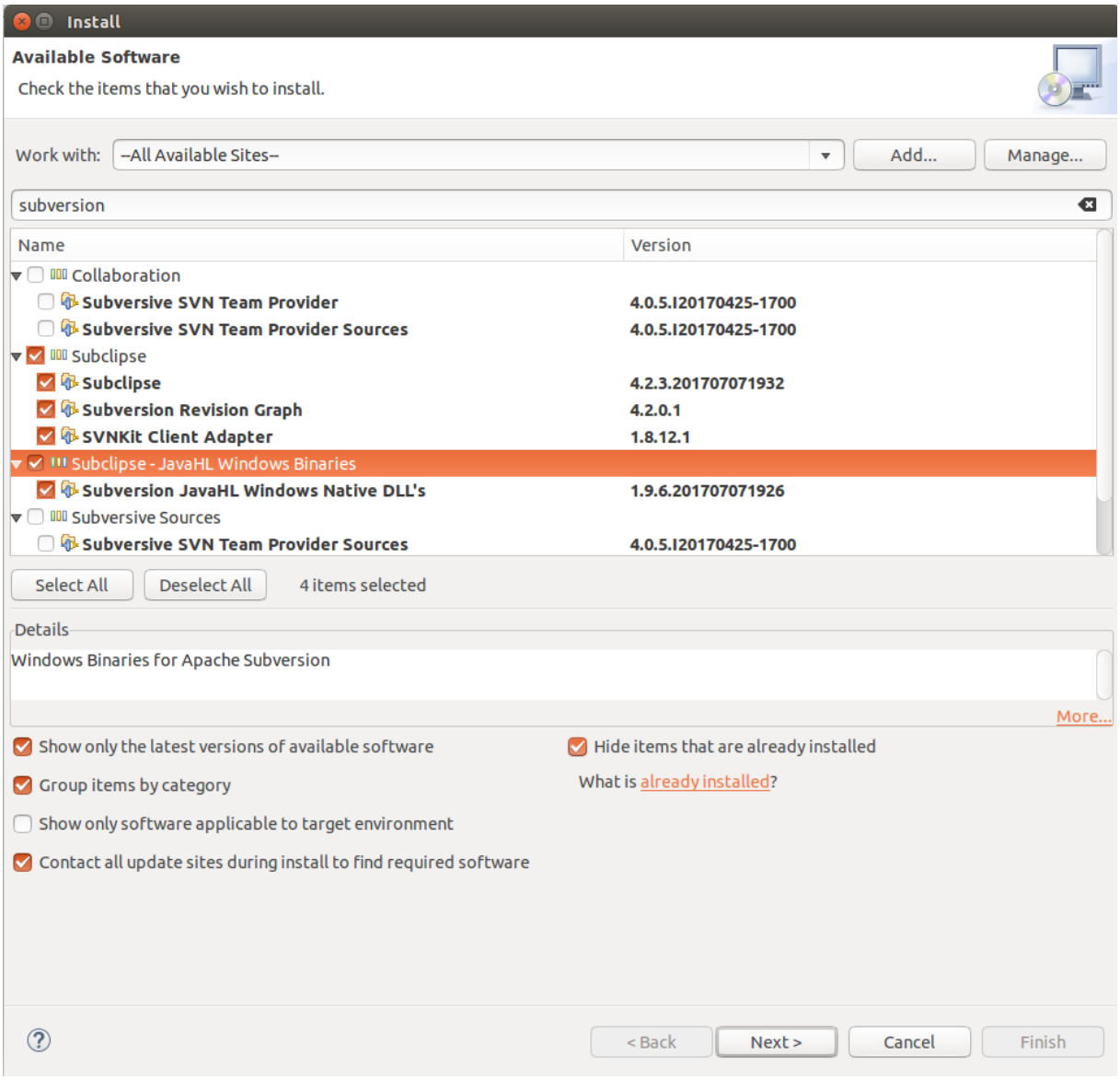
- Navigate to **Help** –> **Install New Software...**



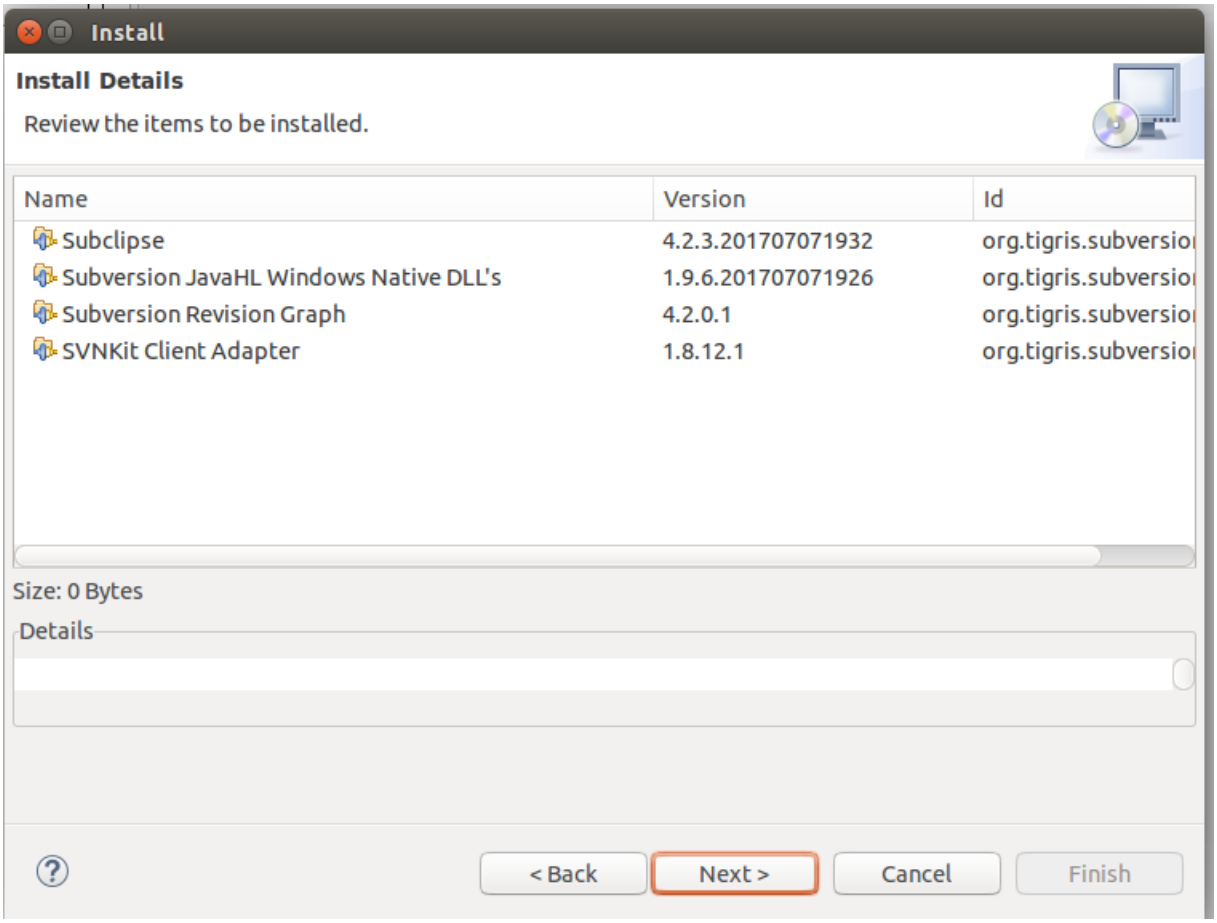
- Change work with section to **All available websites**



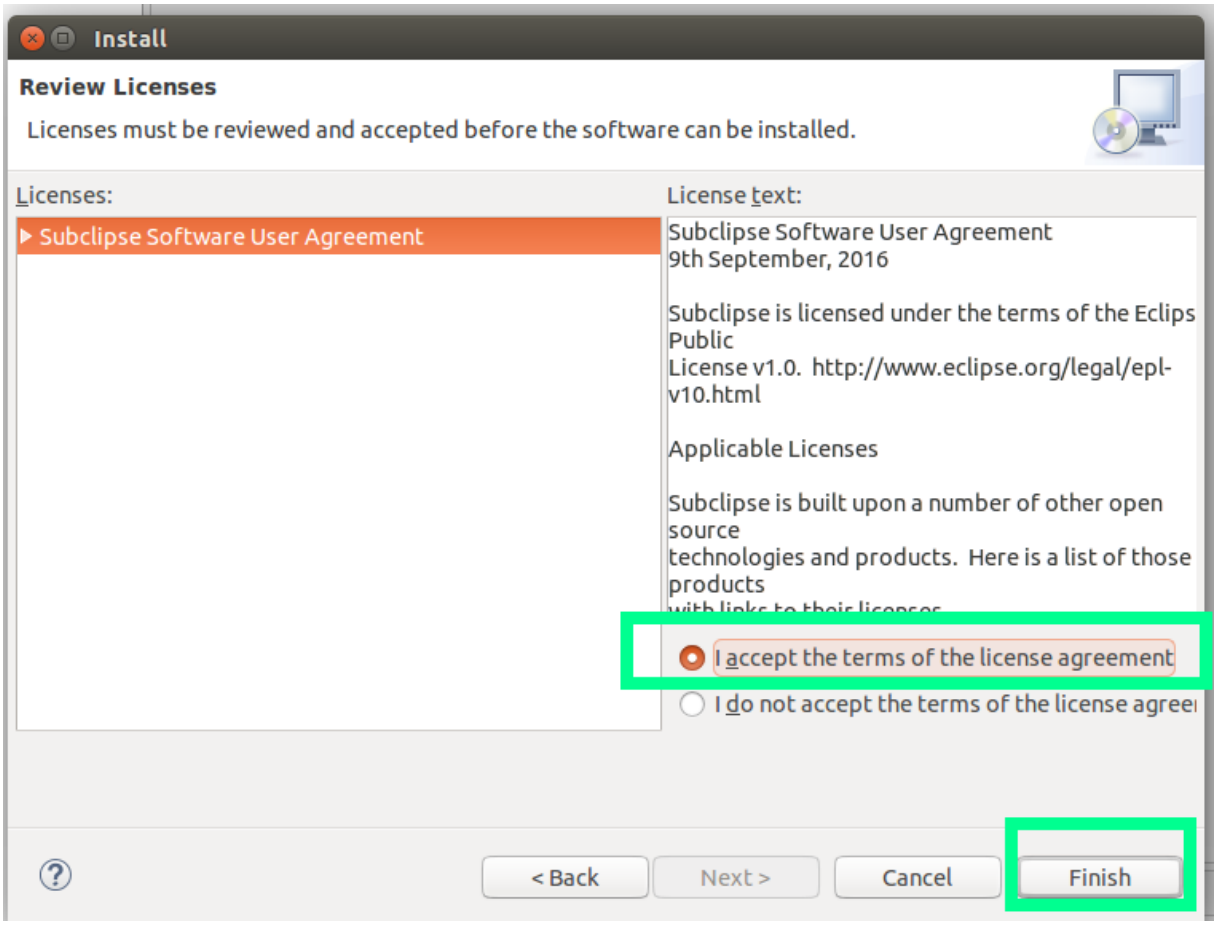
- Write **subversion** to searching bar and select only **Subclipse** and **Subclipse - JavaHL Windows Binaries**



- Continue with the next



- Proceed as shown below



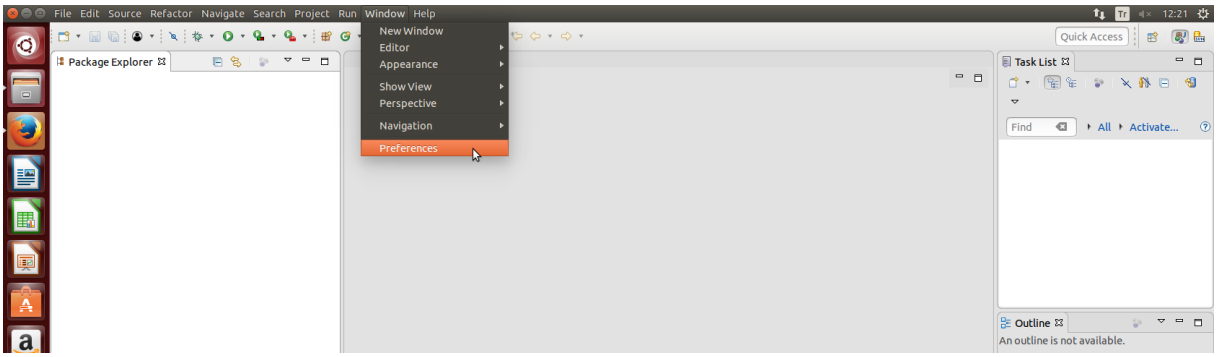
If you get a security warning just continue with -Install Anyway- option

Once you are done with your installation restart your eclipse IDE again.

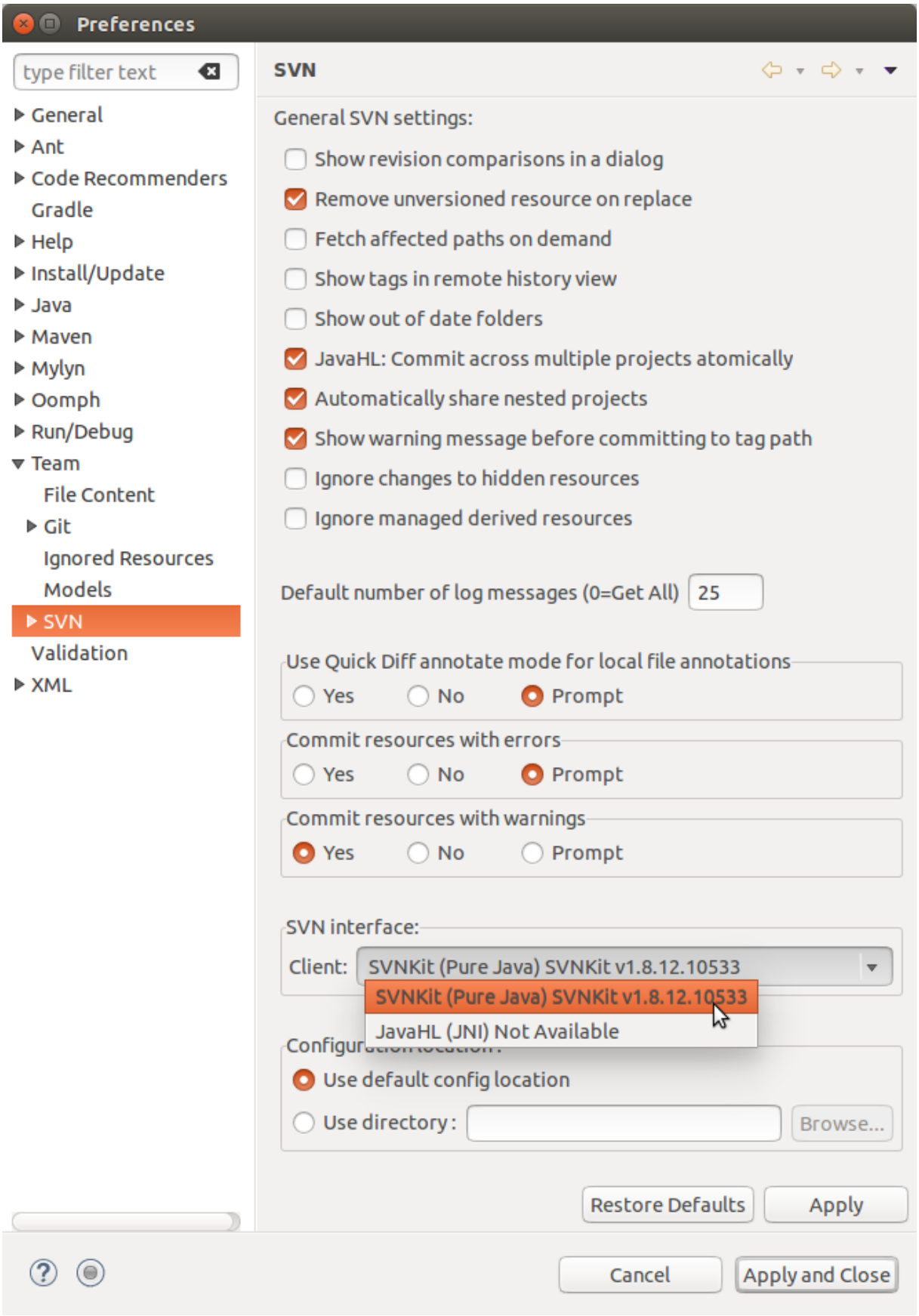
As for last step before we proceed with importing the svn files we should set up our **SVN INTERFACE**

To do that,

- Click on **Window –> Preferences** in the Eclipse IDE



- Navigate to **Team —> SVN** and on the right hand side you might find svn interface set to " JAVAHL(JNI) Not Available. Change it to “SVNKit”

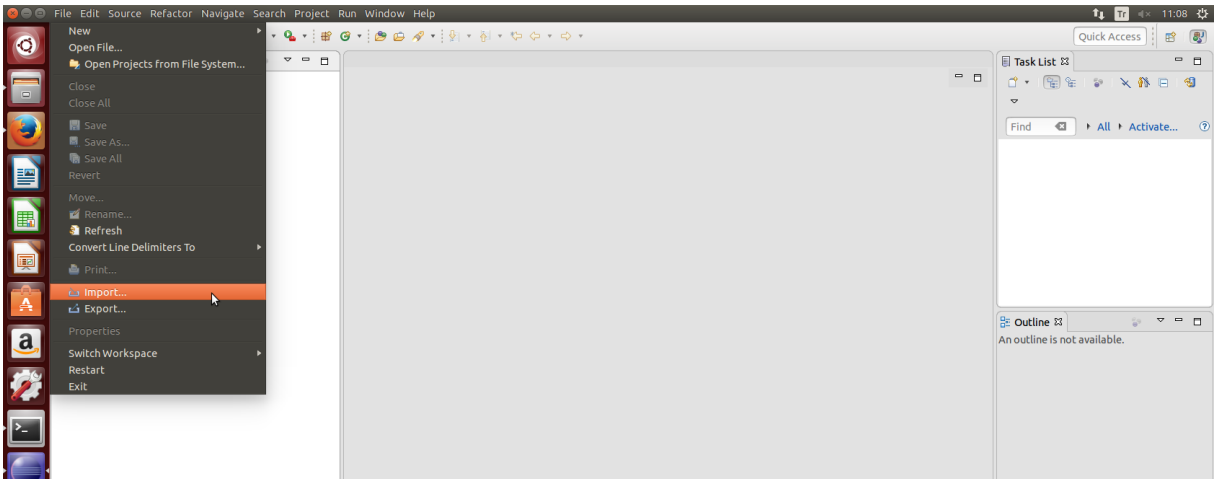


Now we are ready to go and install svn files from the svn server of the University of Konstanz.

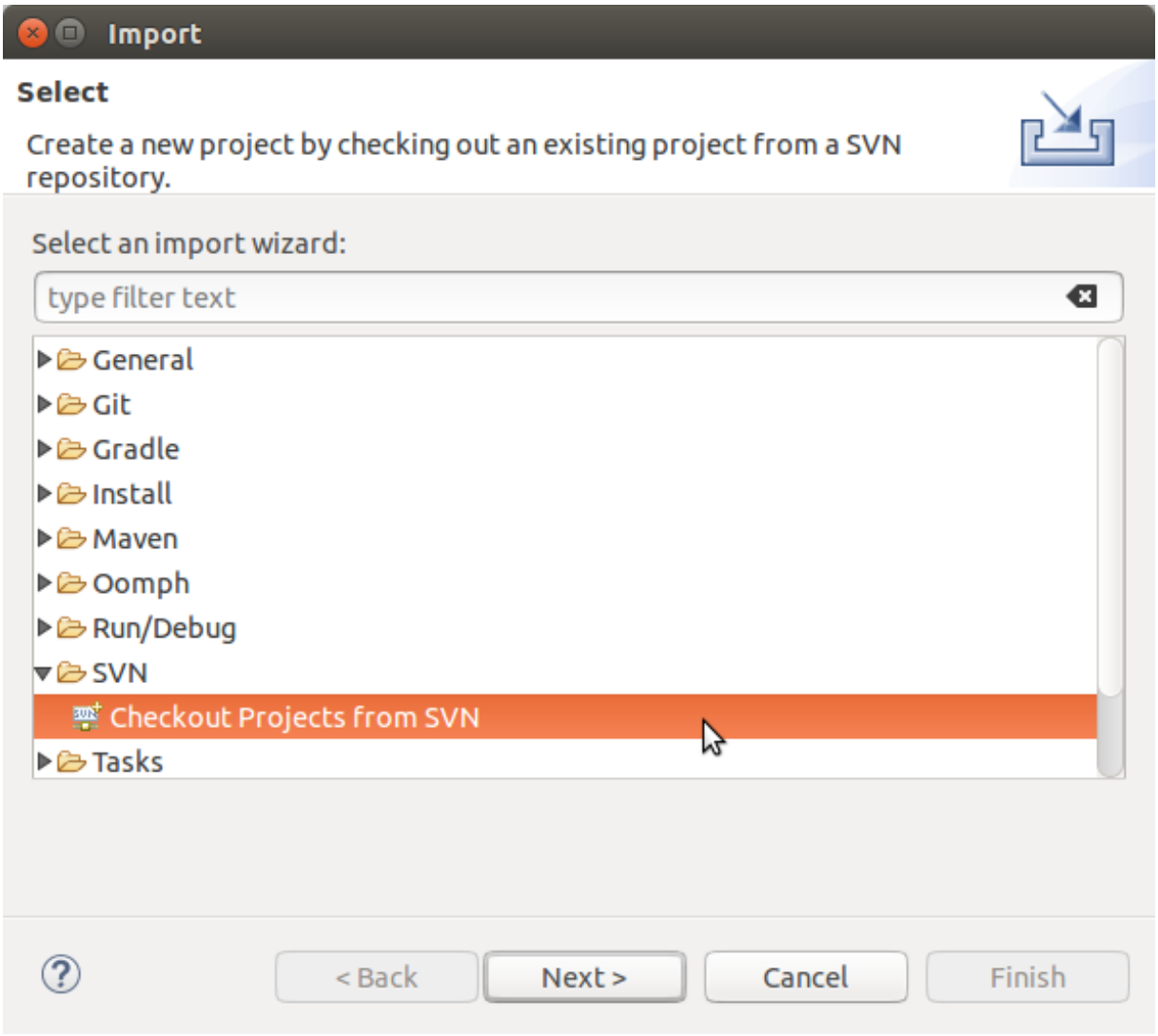
3 Importing the Source Code from SVN Server of the University of Konstanz

Open Eclipse again and jump into Workbench.
Now that we have installed the subversion, we should download the svn files respectively.

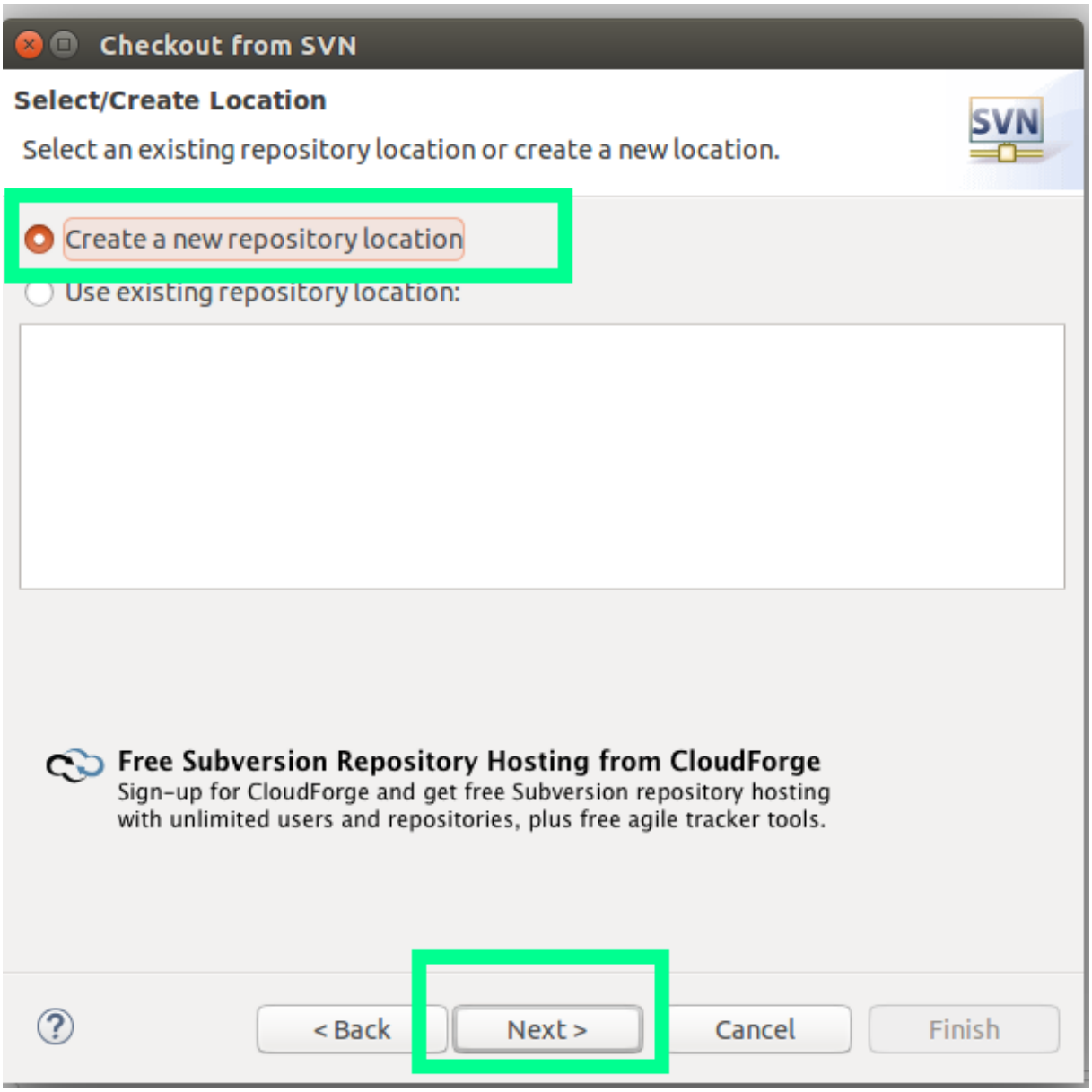
- Go and click the import from **File** –> **Import**



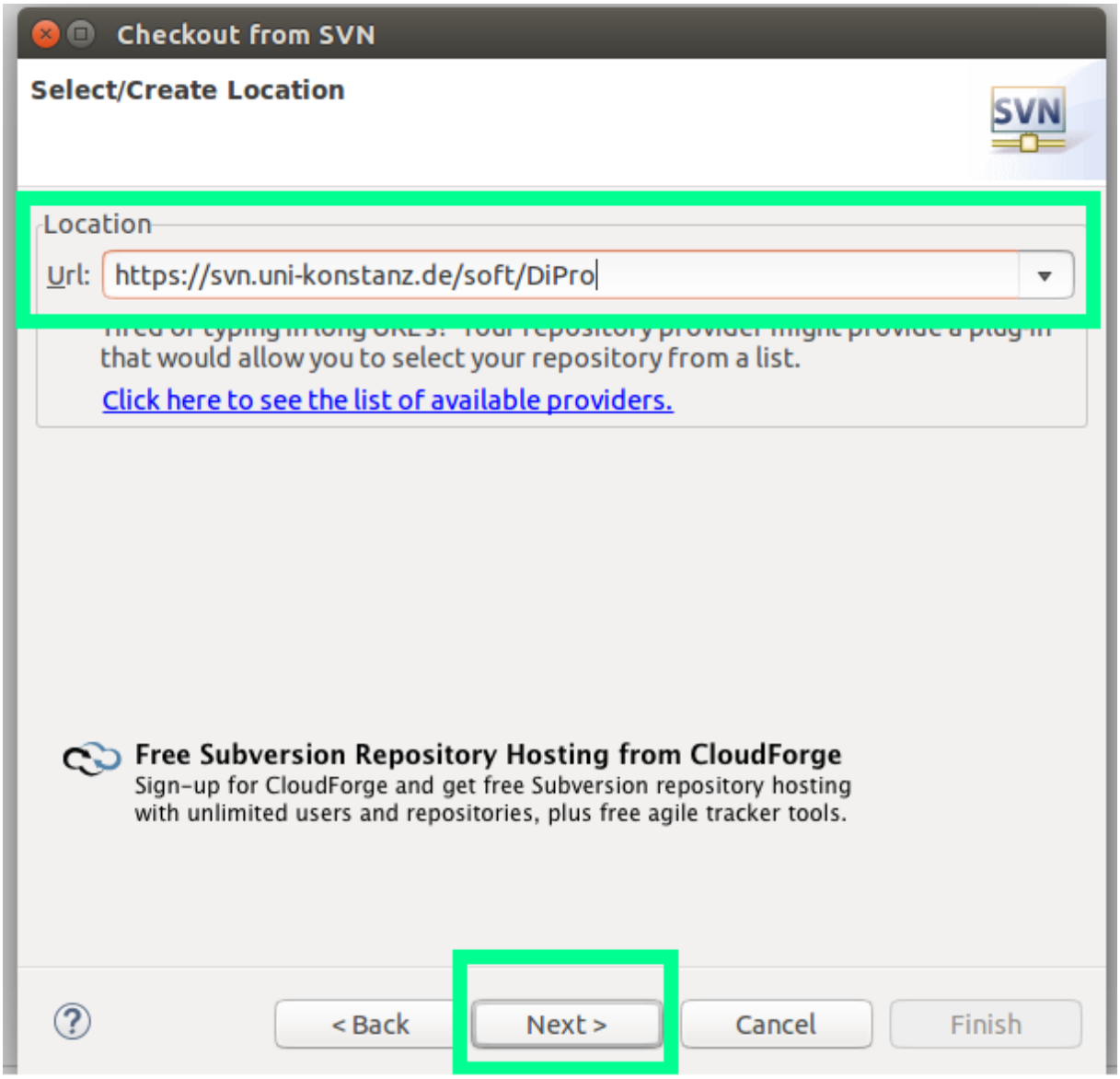
- On **import** window , select **SVN** and **Checkout Projects from SVN** and click **next**



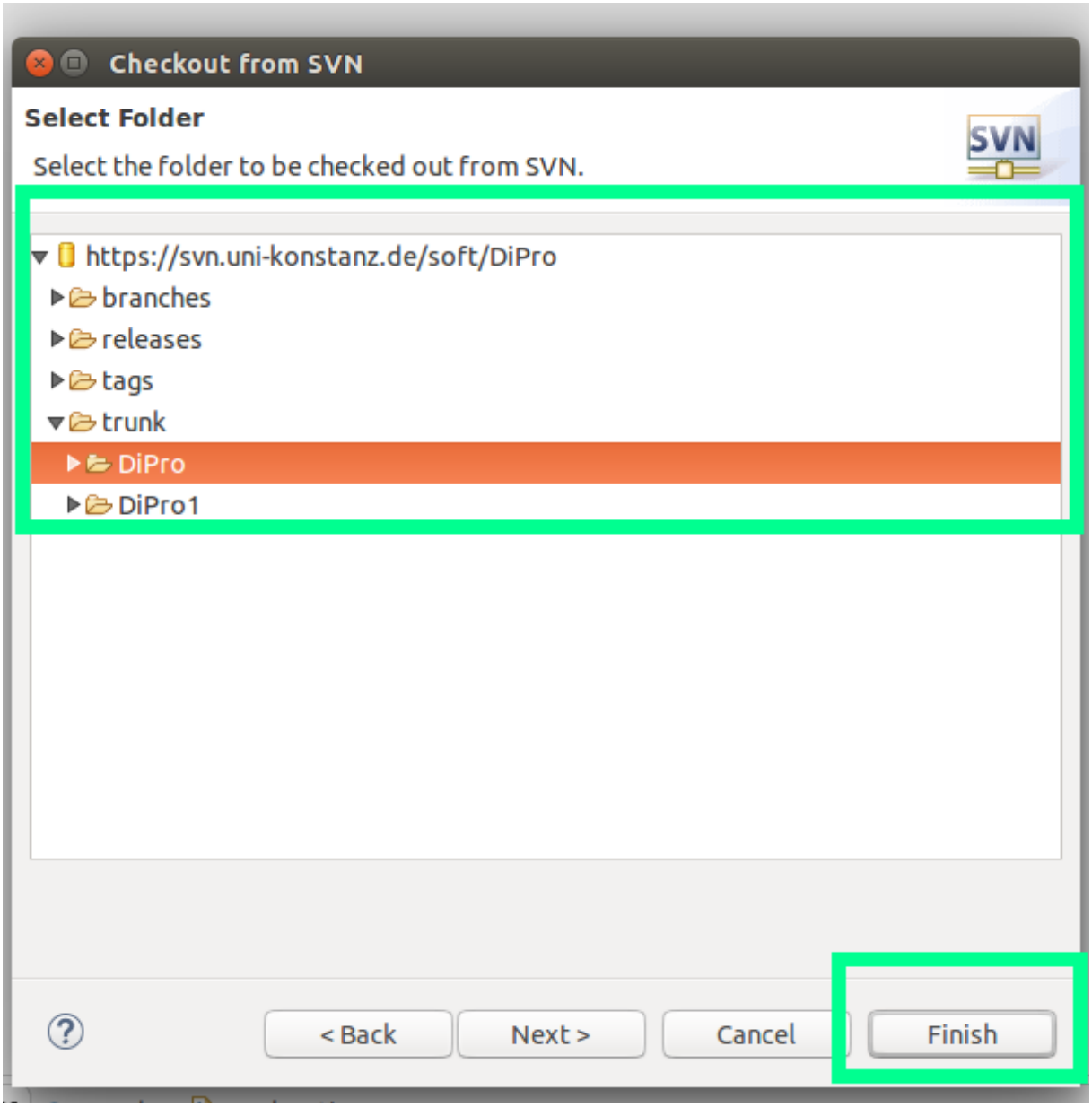
- Select **Create a new Repository Location**



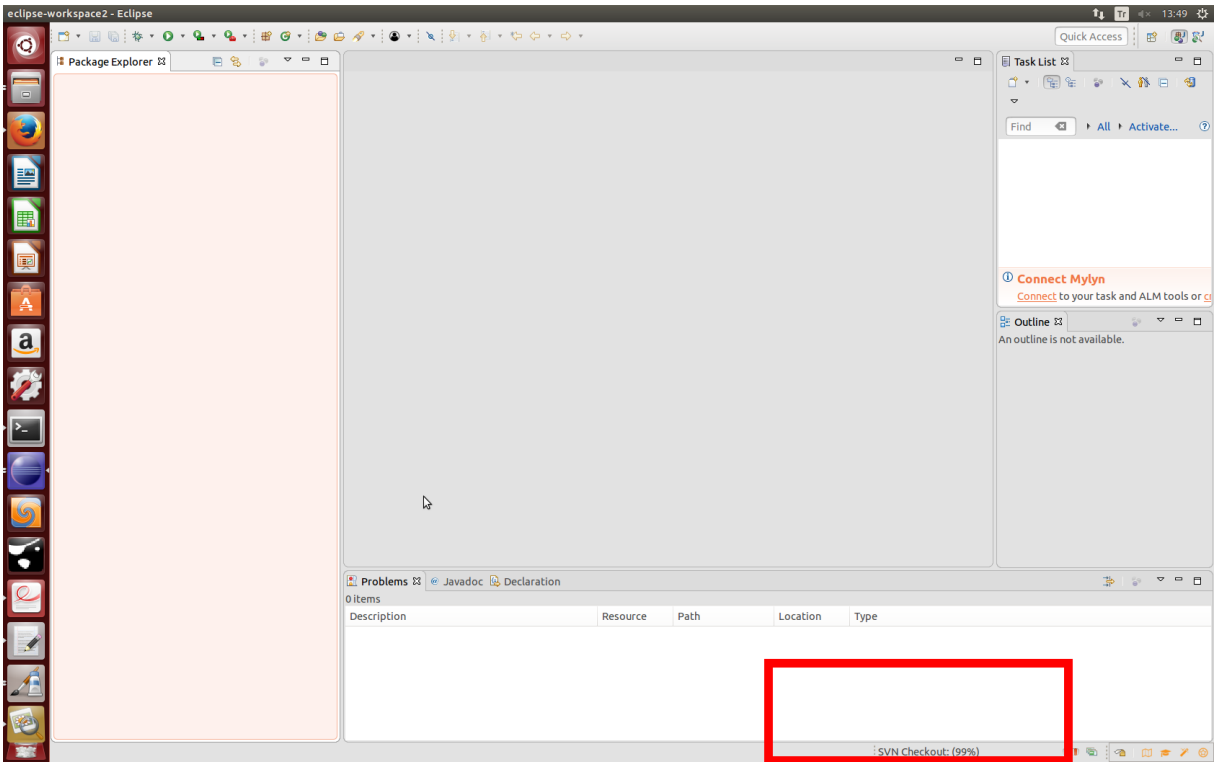
- Use **https://svn.uni-konstanz.de/soft/DiPro** address to checkout from svn file and click yes



- Make sure to Expand **Svn Server** —> **trunk** —>**DiPro** and hit **Finish**

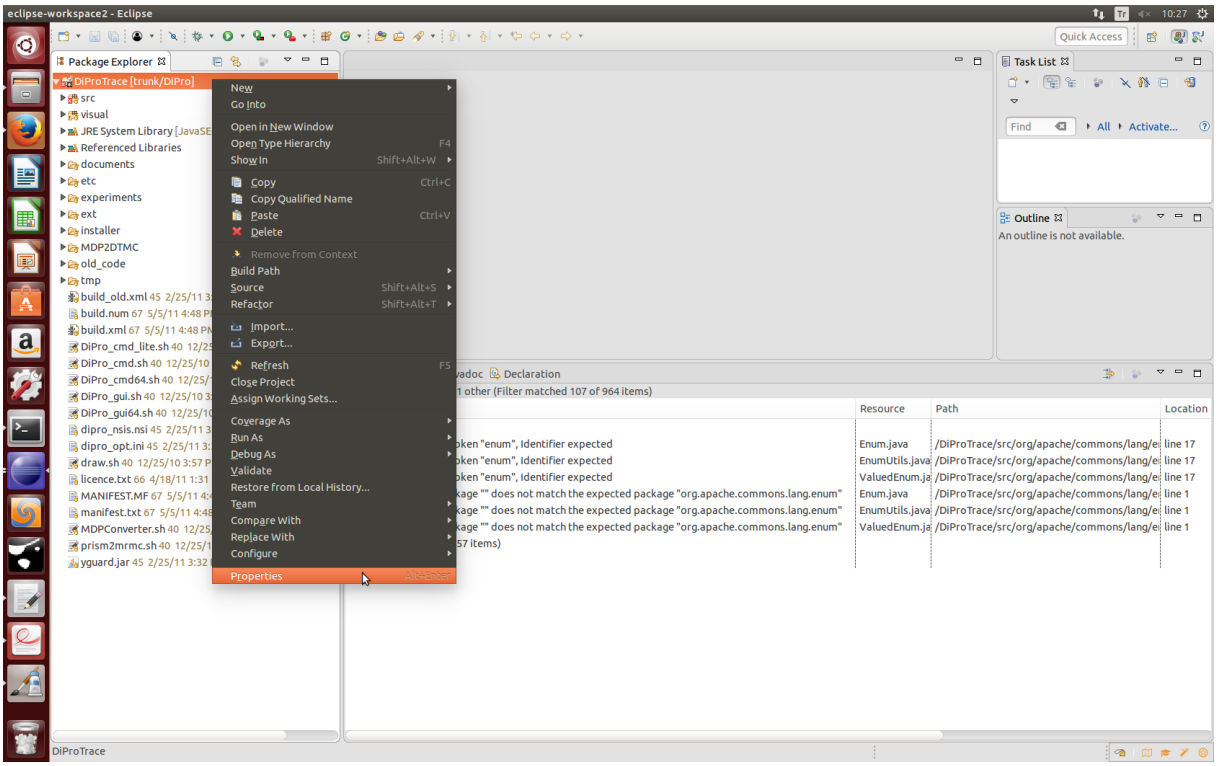


- Download fro Svn server will start as you hit on **Finish** button, you can view the progress on the bottom right-hand side.

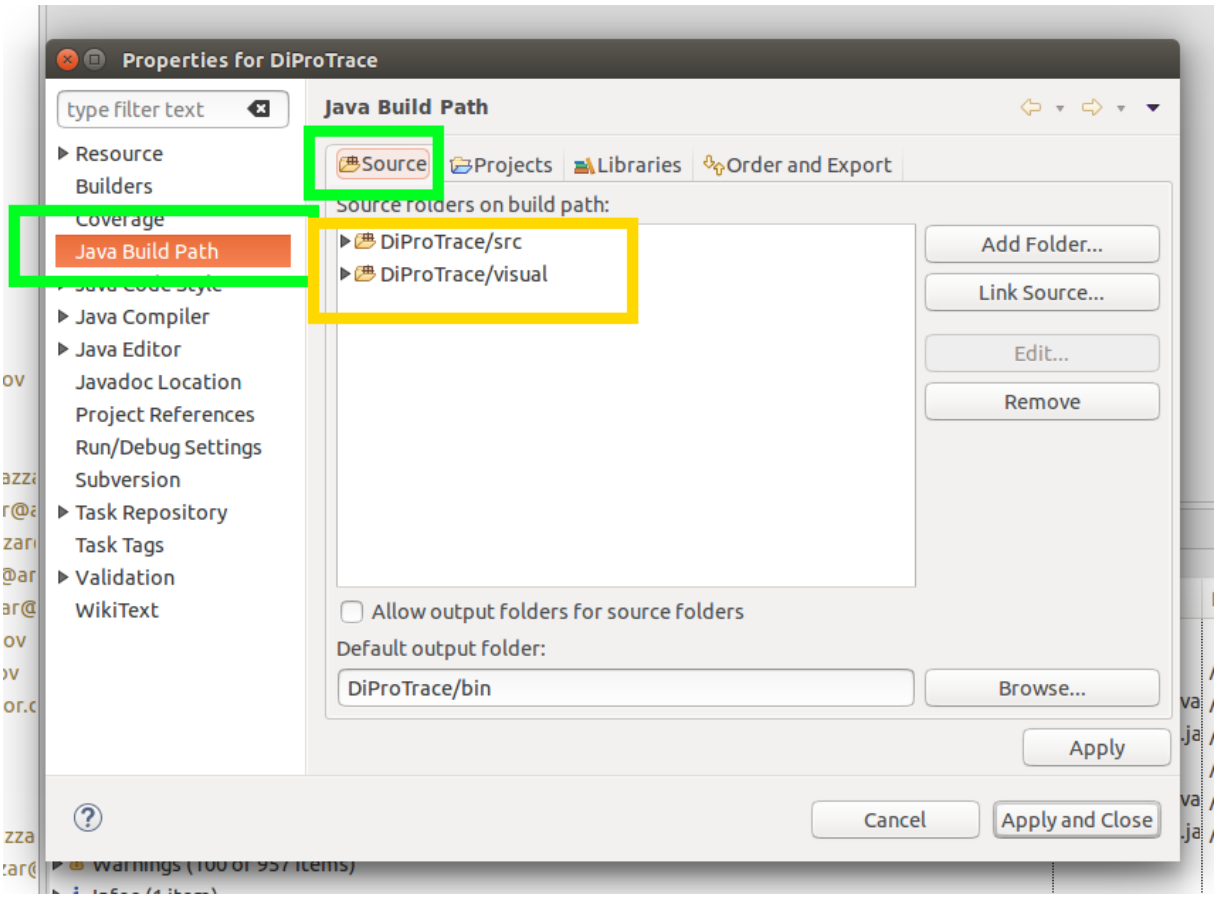


4 Modifying DiPro Project source folder for correct configuration

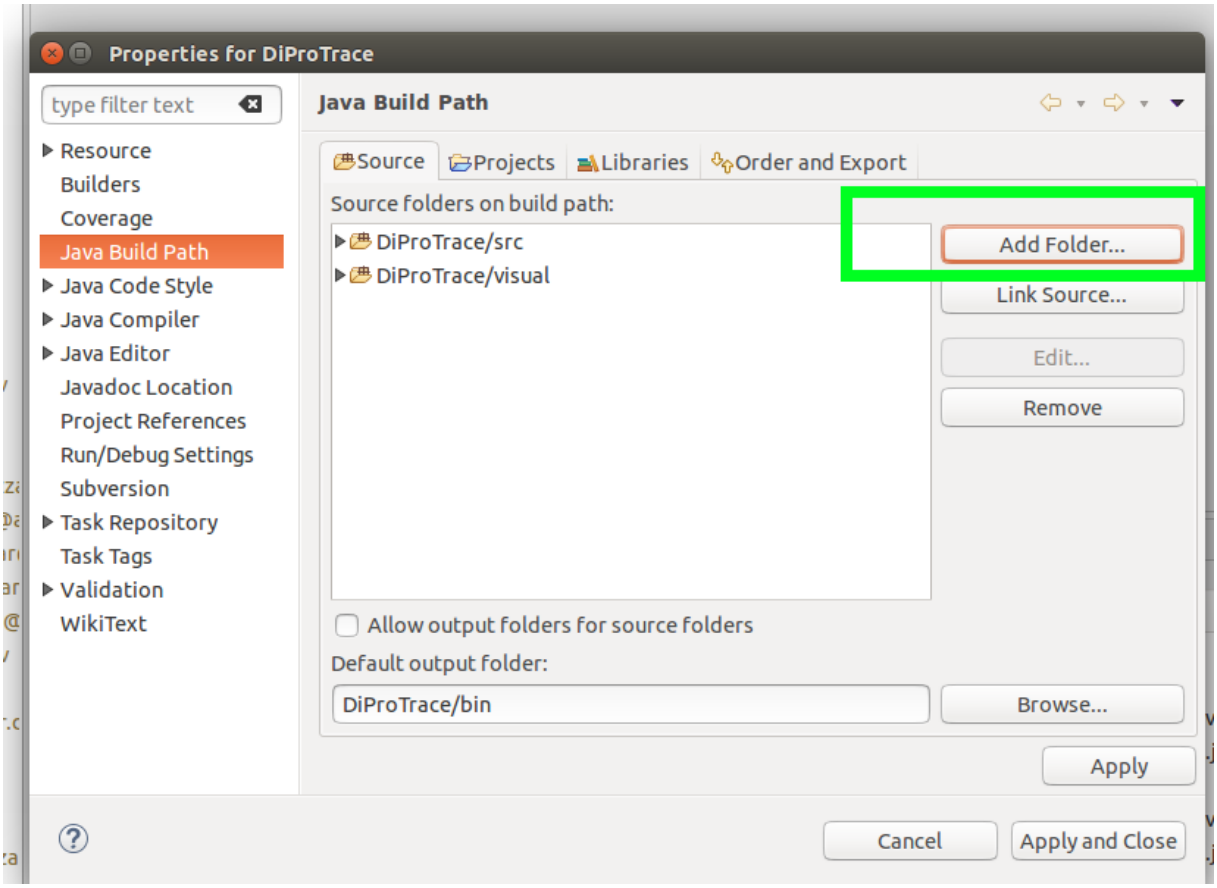
As we solved our errors now we must include our etc subfile to our project source folder in order to let it use required images(such as splash image of the dipro while opening) while using the visual gui for dipro, to do that navigate to **project explorer view** and right click on project and select **Properties**



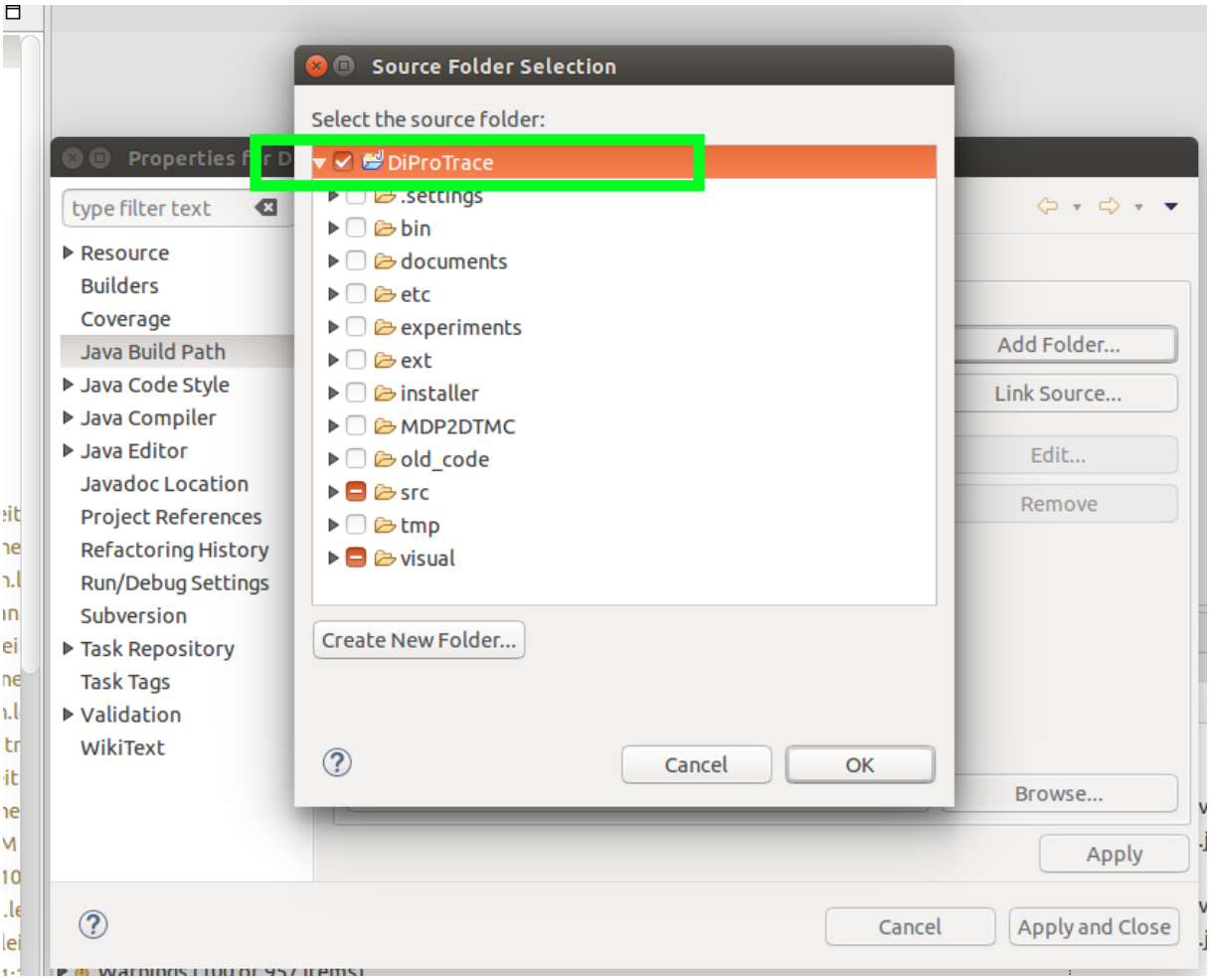
- Now navigate to **Java Build Path** and come to **Source Section** , folders marked with yellow should be modified to clean errors from the project.



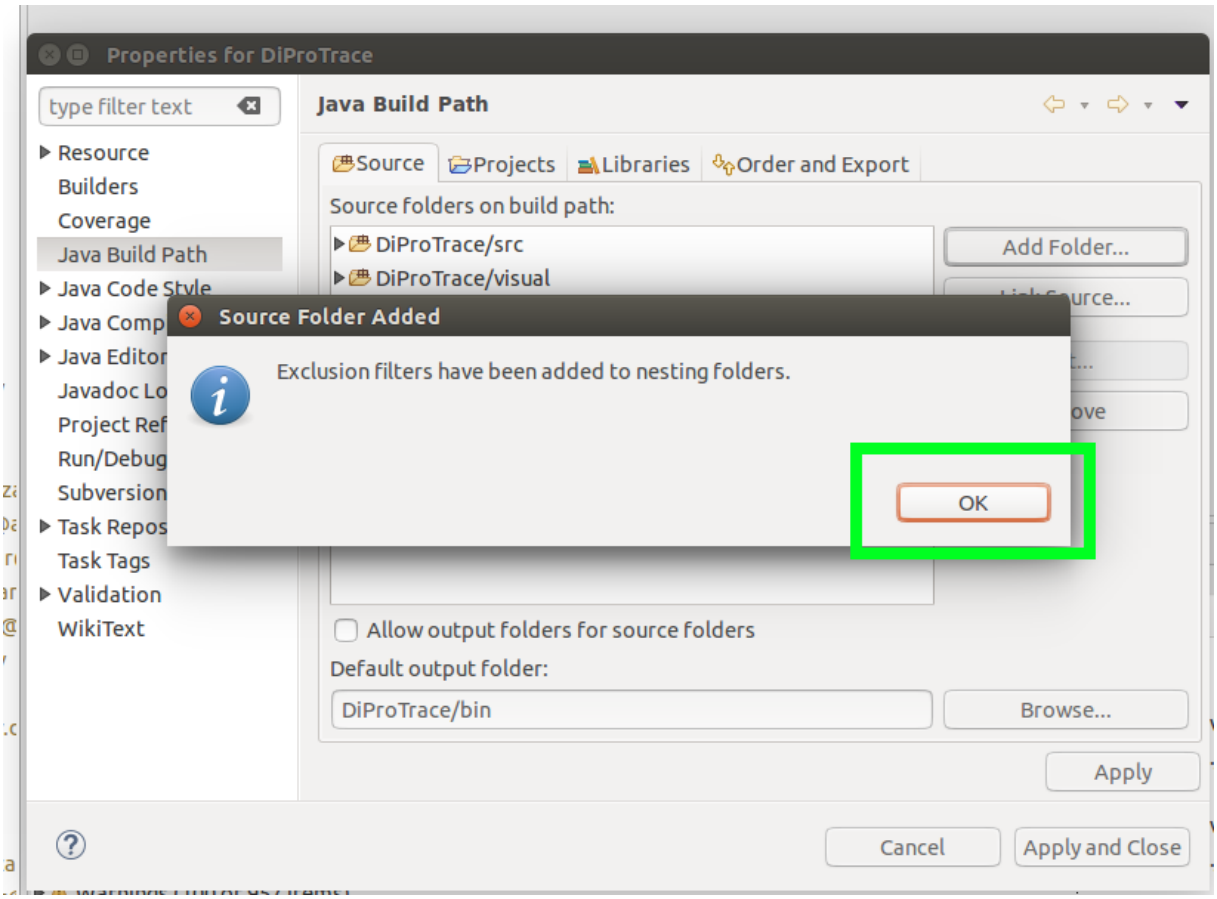
- Click on **Add Folder**



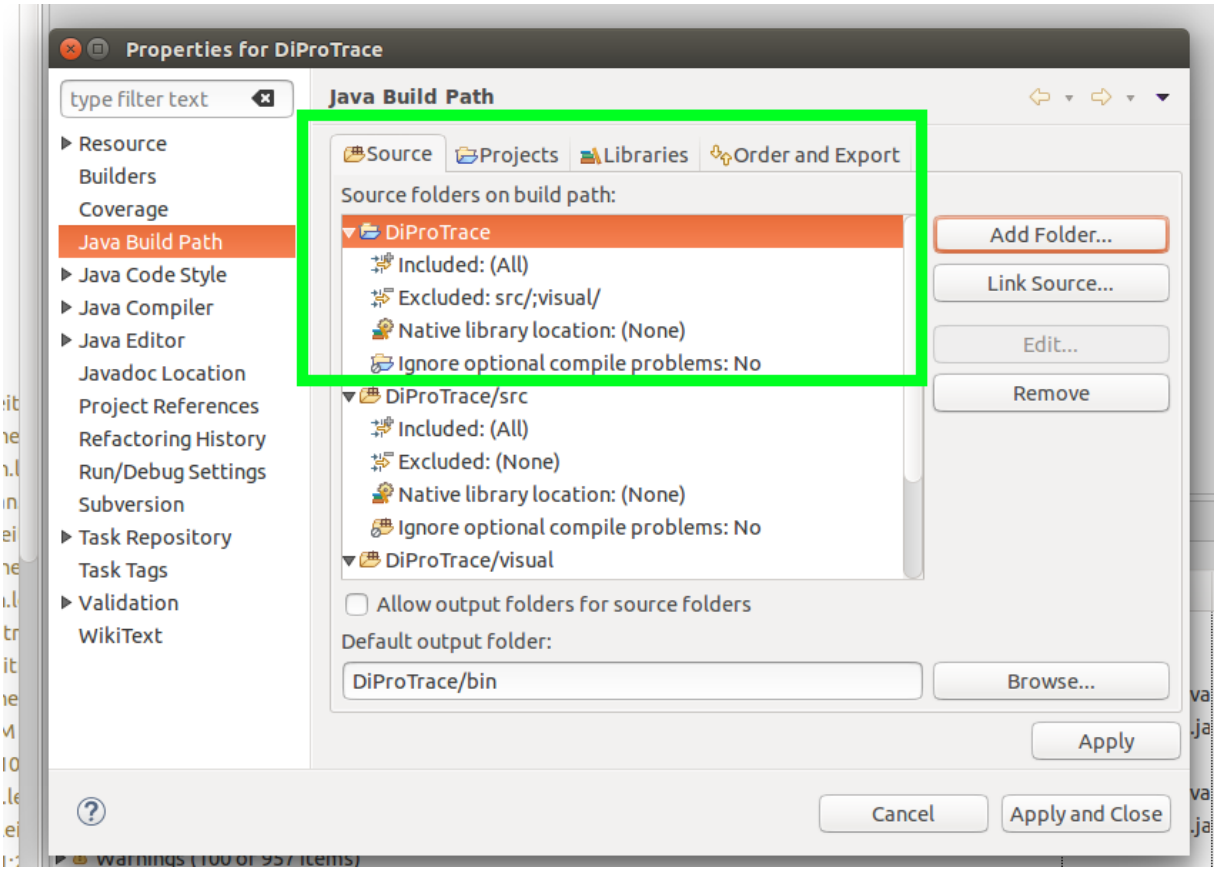
- Initially **DiProTrace** is unchecked, so check the checkbox left next to it and click on **okey**



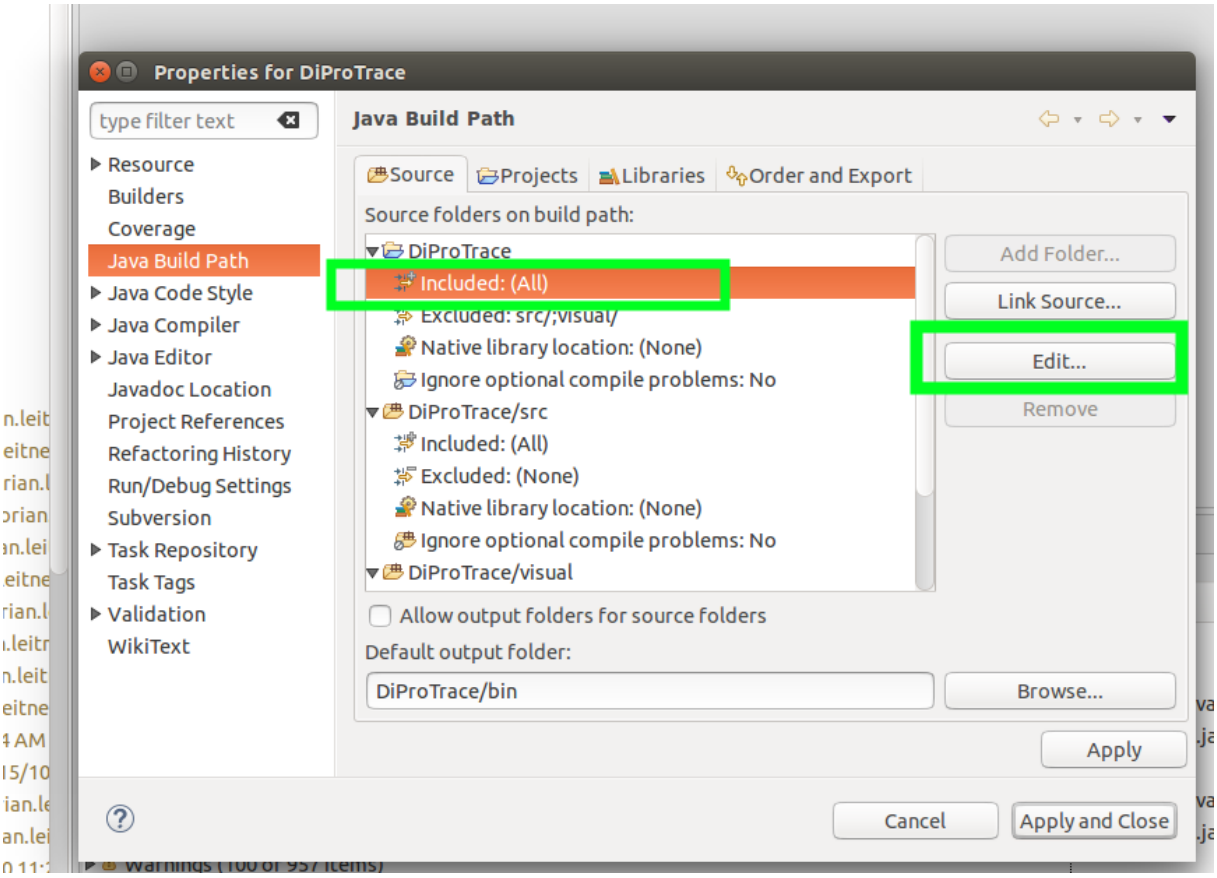
- There is going to be informing view about folder source folder being added, just click on **okey**



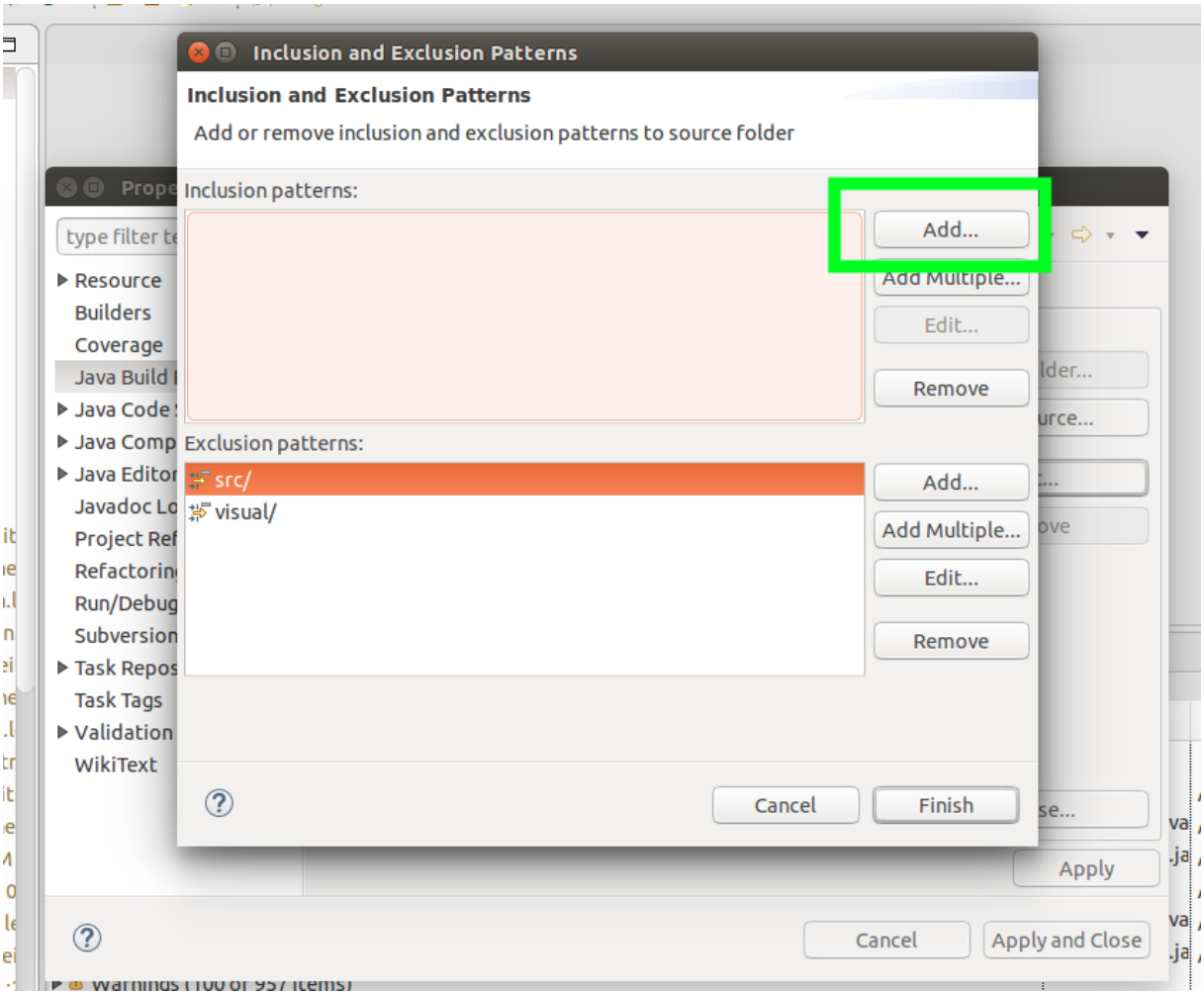
- Once it is added we need to slightly modify it.



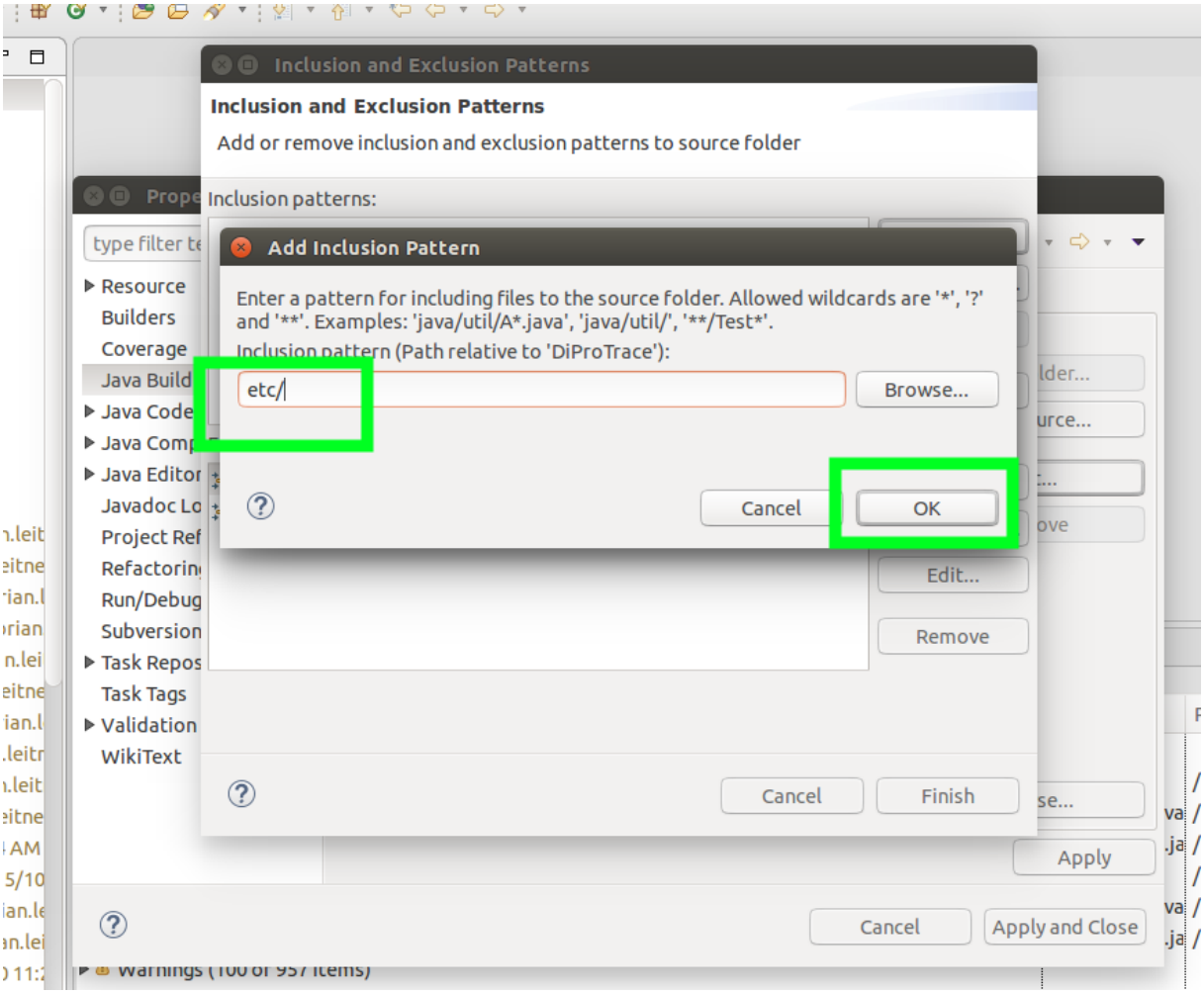
- Make sure that it is like above and click on **Included** and then hit on **Edit**.



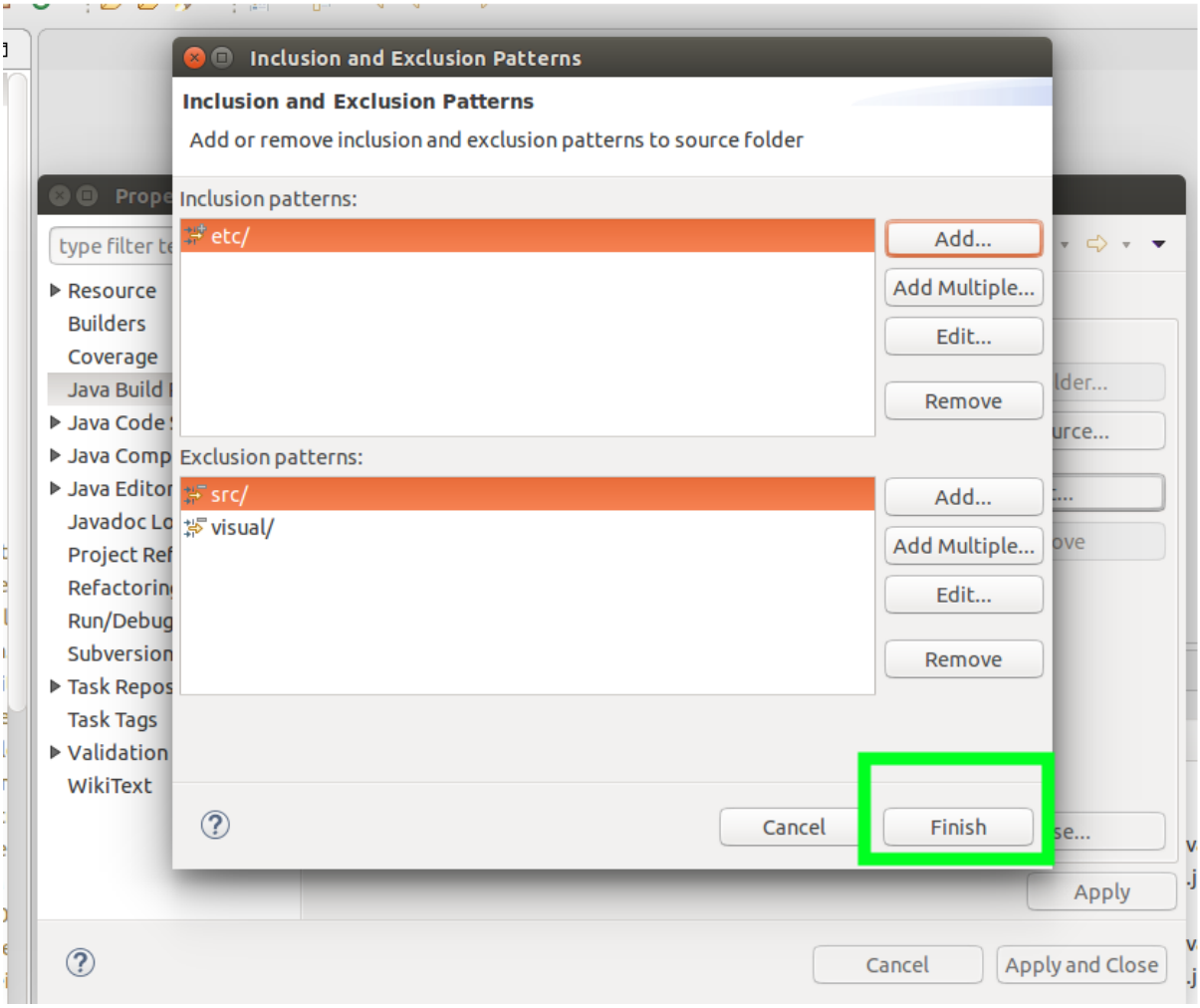
- Click on Add for adding a new inclusion pattern.



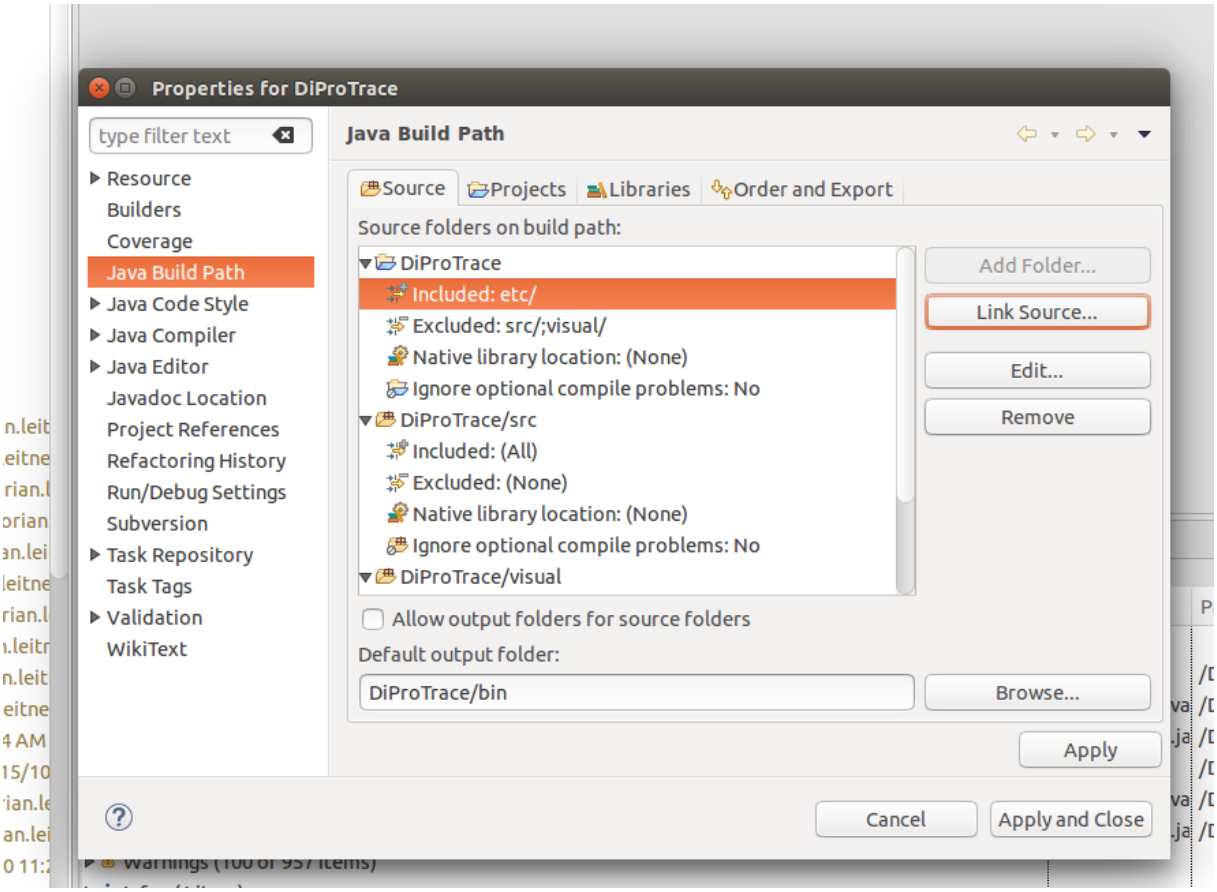
- Type **etc/** and click on **okey**



- Now that you added the etc subfolder you can click on **finish**



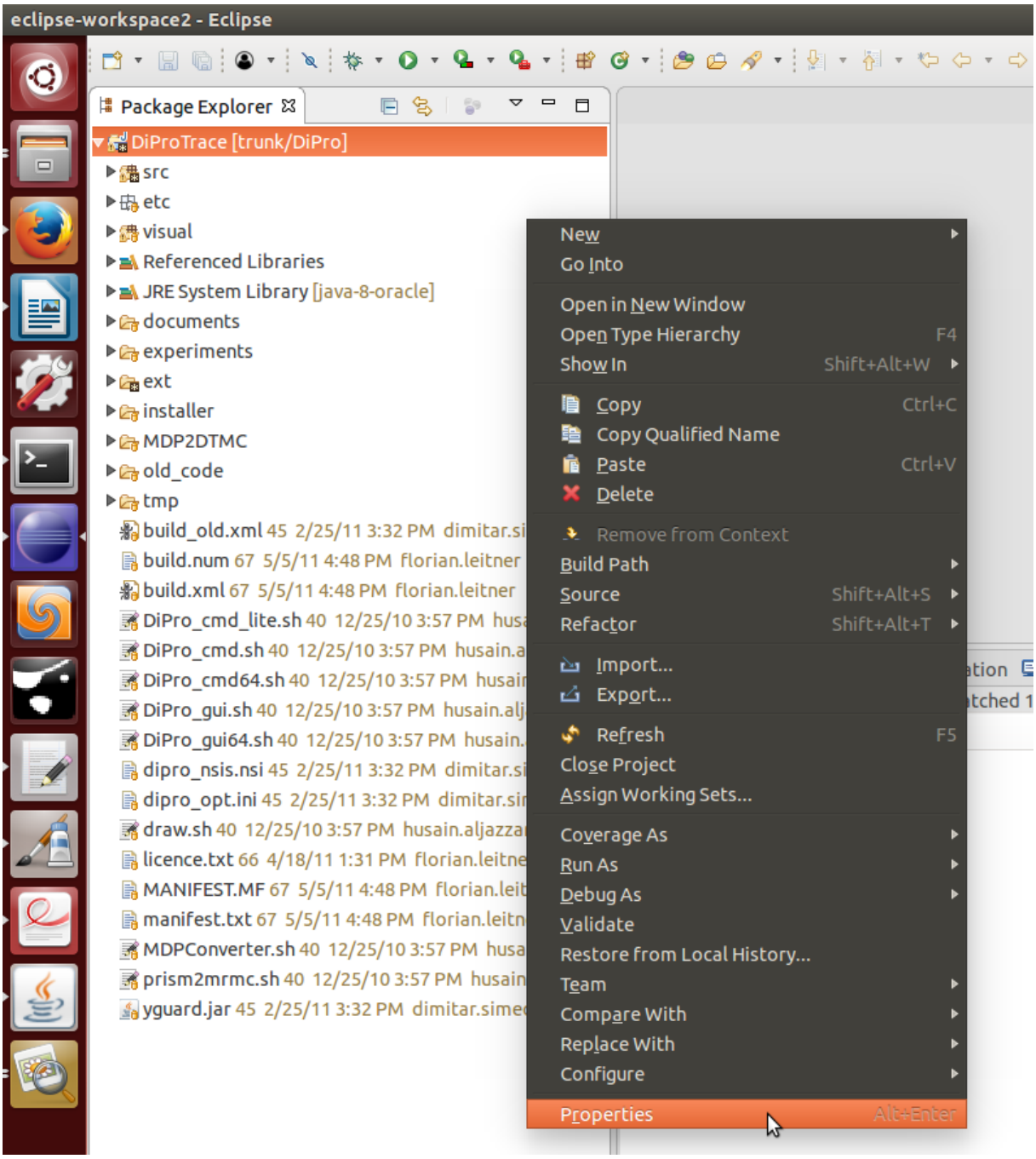
- As we've finished our modifying, final view should be like this, if so, click on **Apply and close**



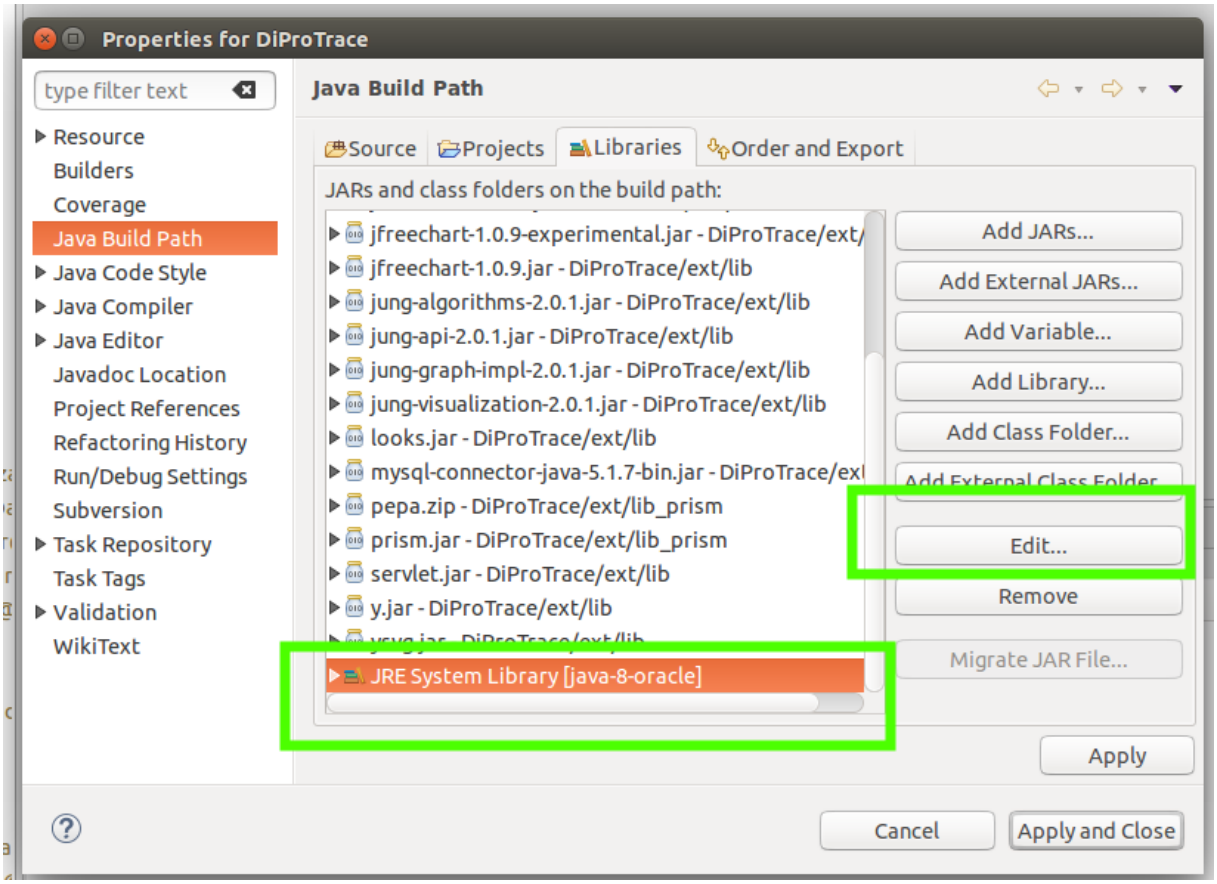
5 Setting Jre System Library to Default(if neccesary)

Sometimes depending on the previous usage of the eclipse its JRE might be set to some other version than normally it is. In order to prevent this problem we can remove and re-add the default JRE library into our project.However, if its already set to default, you can skip this step.

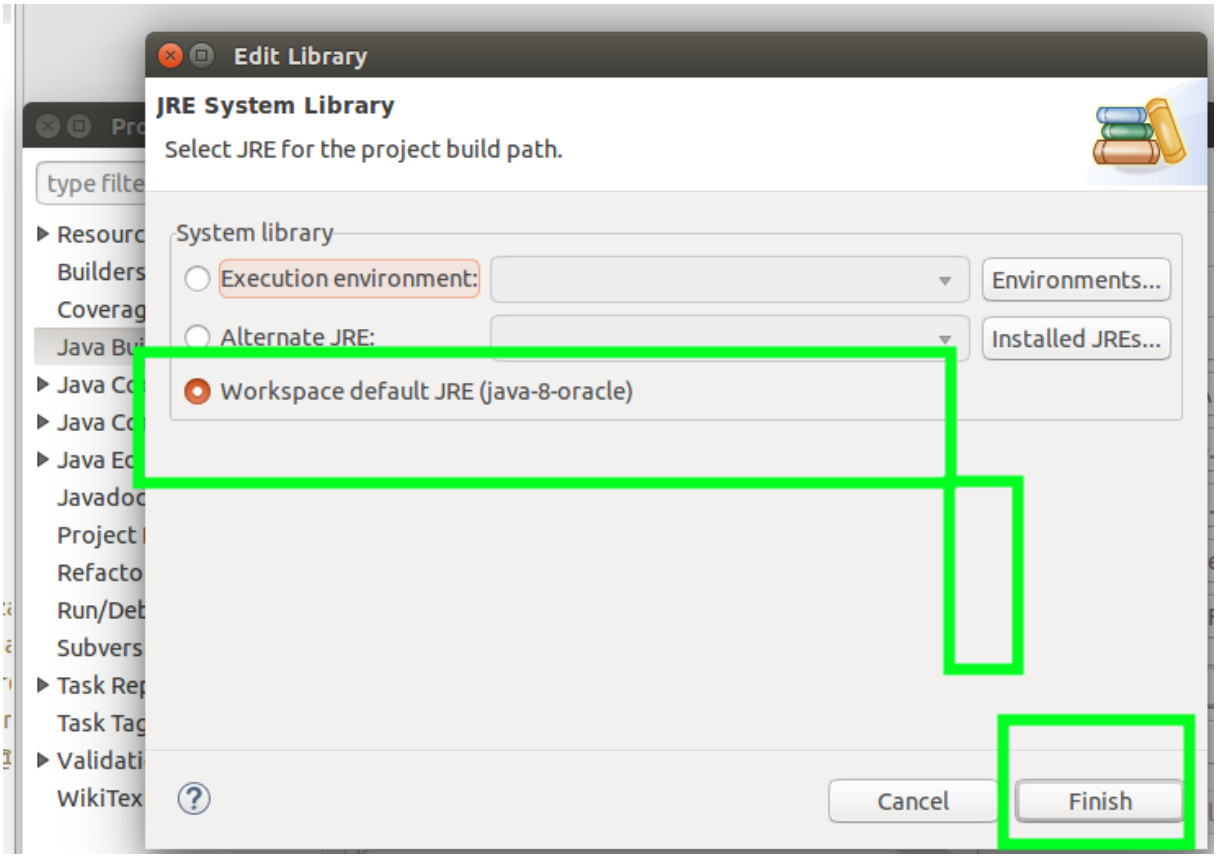
- Right click on the project and select properties.



- Come to **Java Build Path** and select the section **Libraries** and scroll down to navigate **JRE System Library**.



- Click on edit and see the settings.
- Make sure that it is selected as default workspace JRE

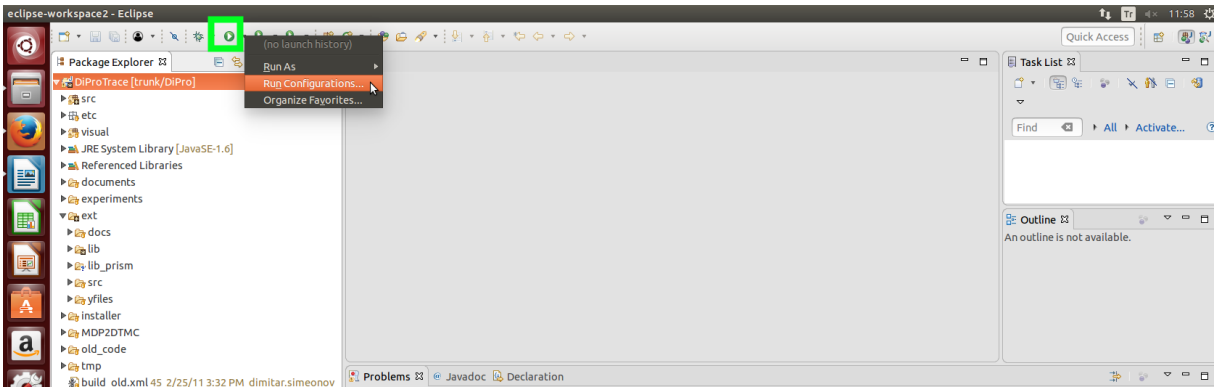


- Click on **Finish**

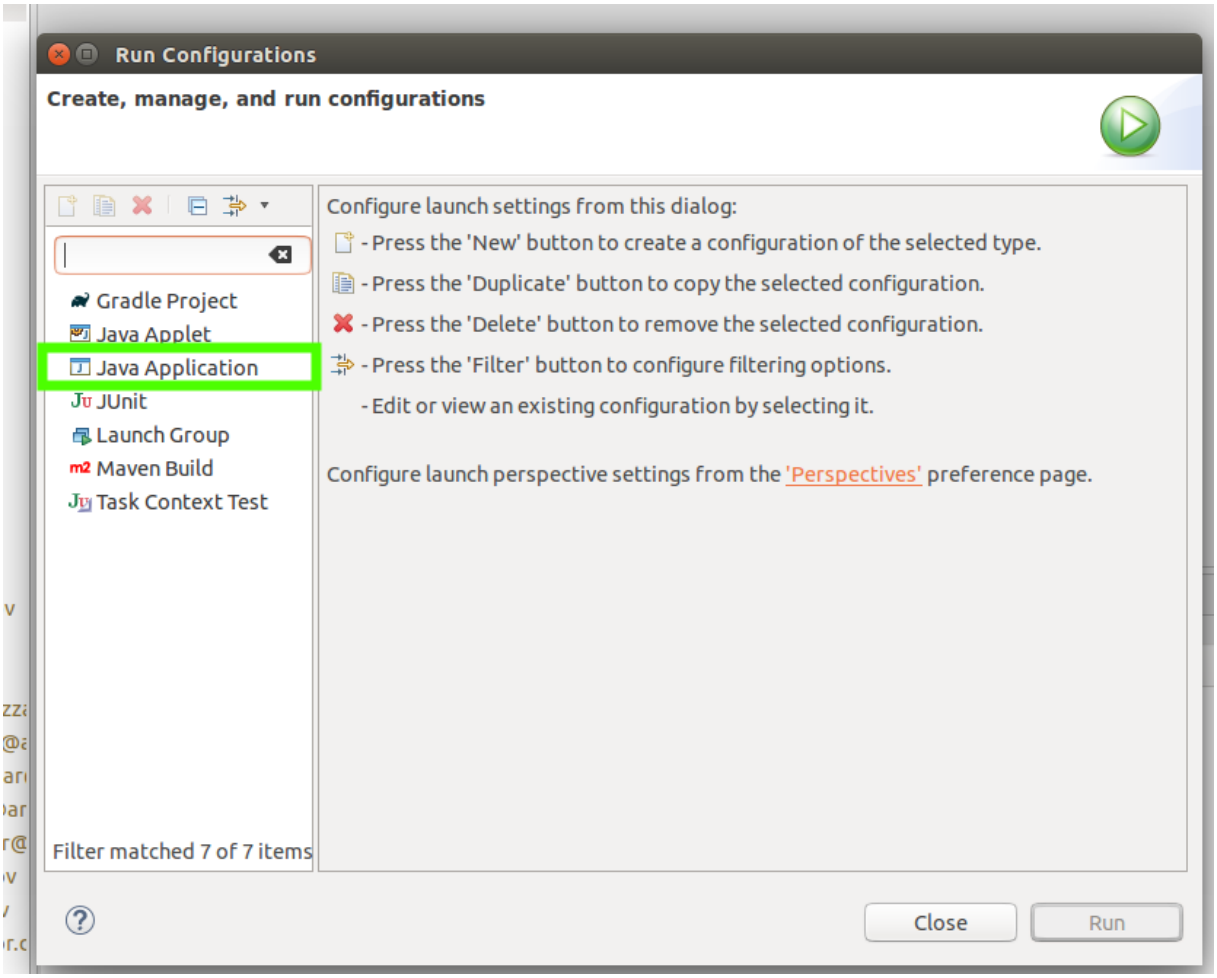
6 How to run DiPro

As we've fixed all errors and modified our project source folder now we are good to go and run DiPro. As a matter of fact, when we run the source code we should modify our running configurations so that everything can go smoothly. To do it, first we should create main and Visual Main. Lets start with creating the console based **main**

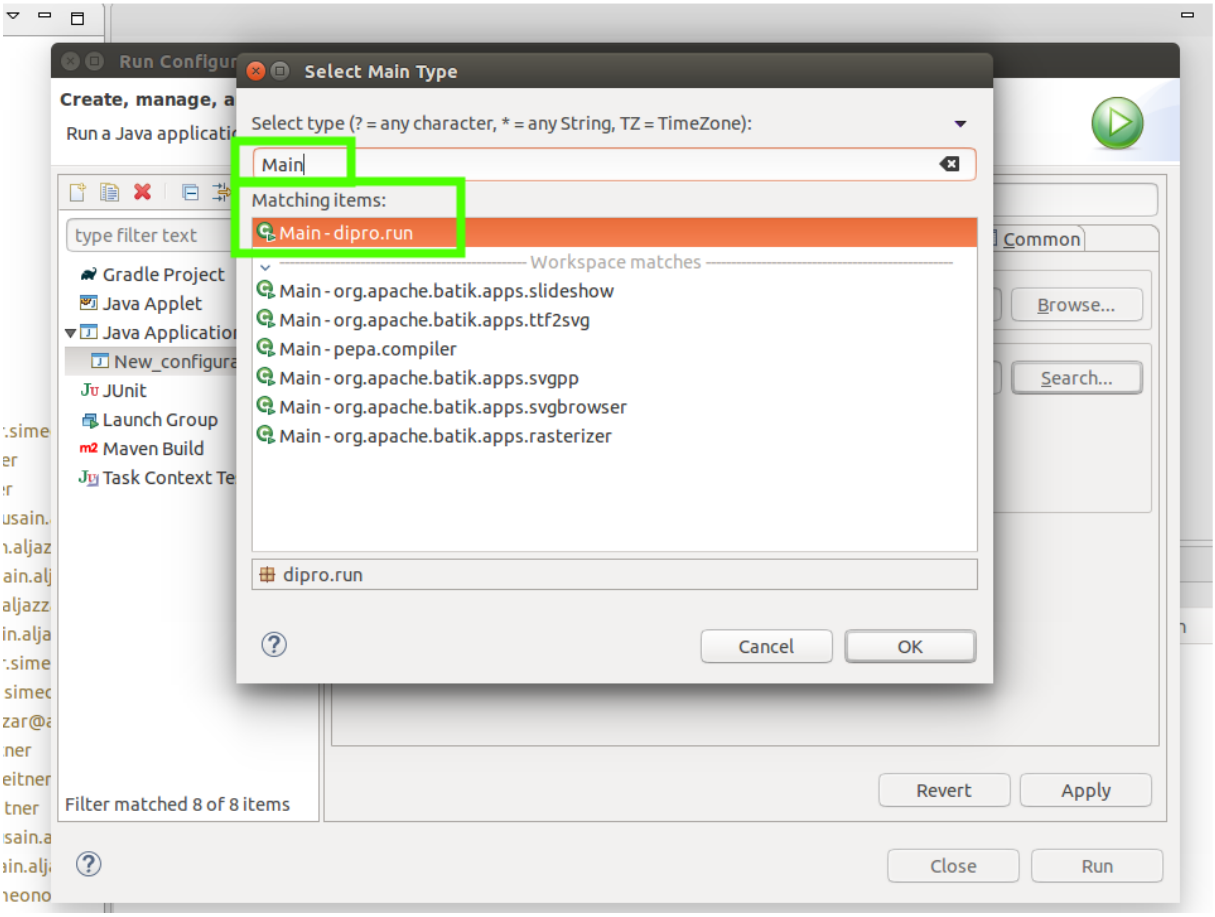
- Click on the small little triangular next to run as and select **Run Configurations.." .



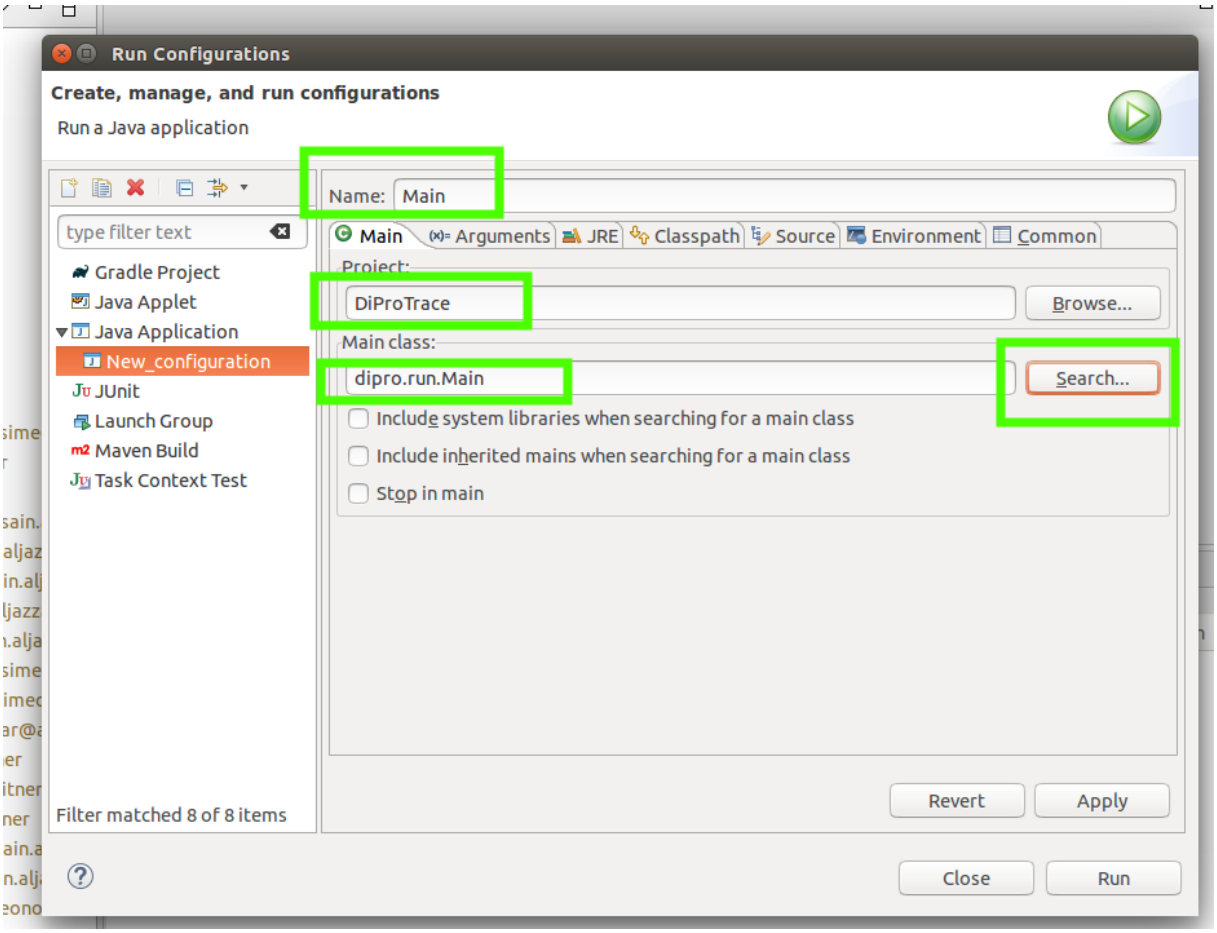
- Double click on the **Java Application** section.



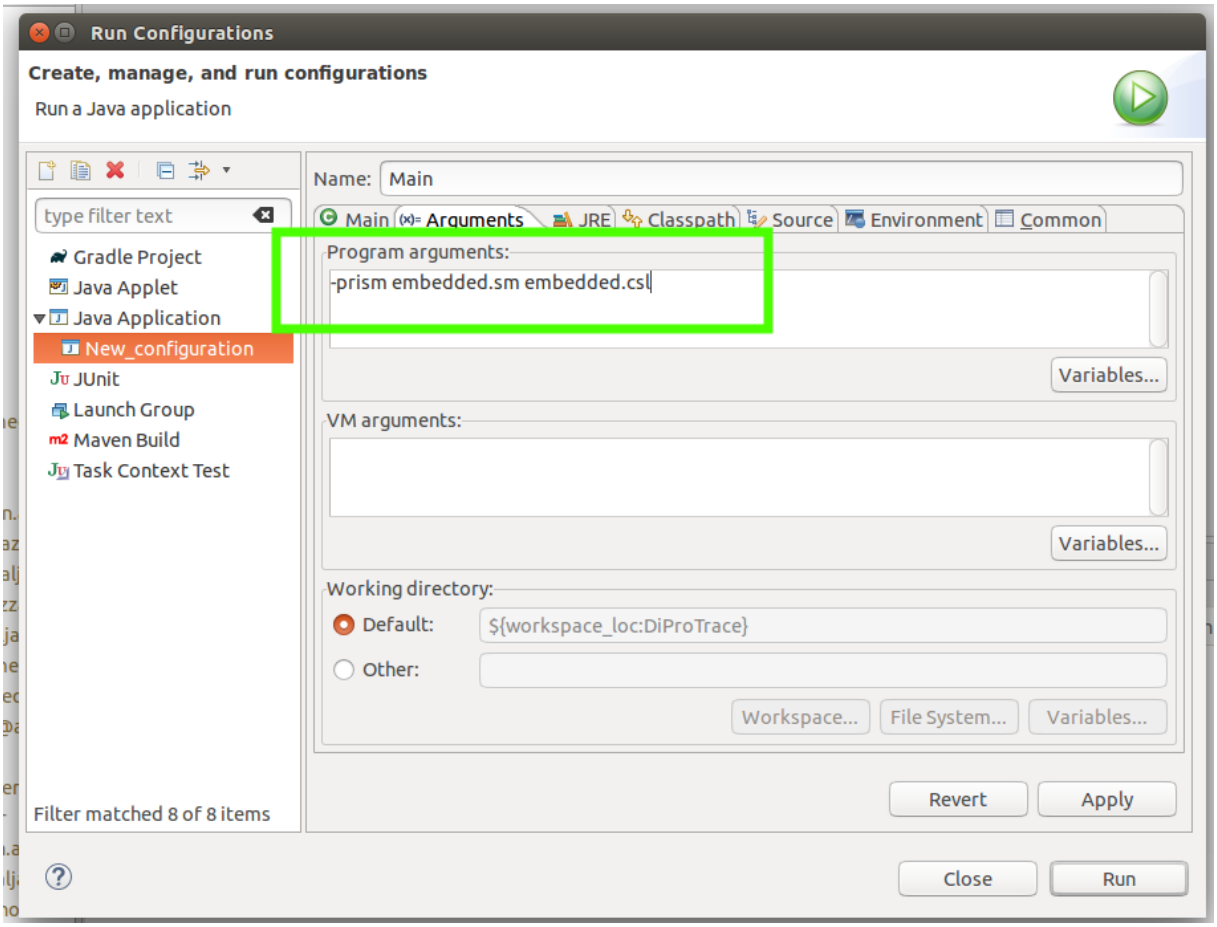
- Name it as **Main** , Project should be selected as **DiProTrace** and below it , click on Search and search for **Main**



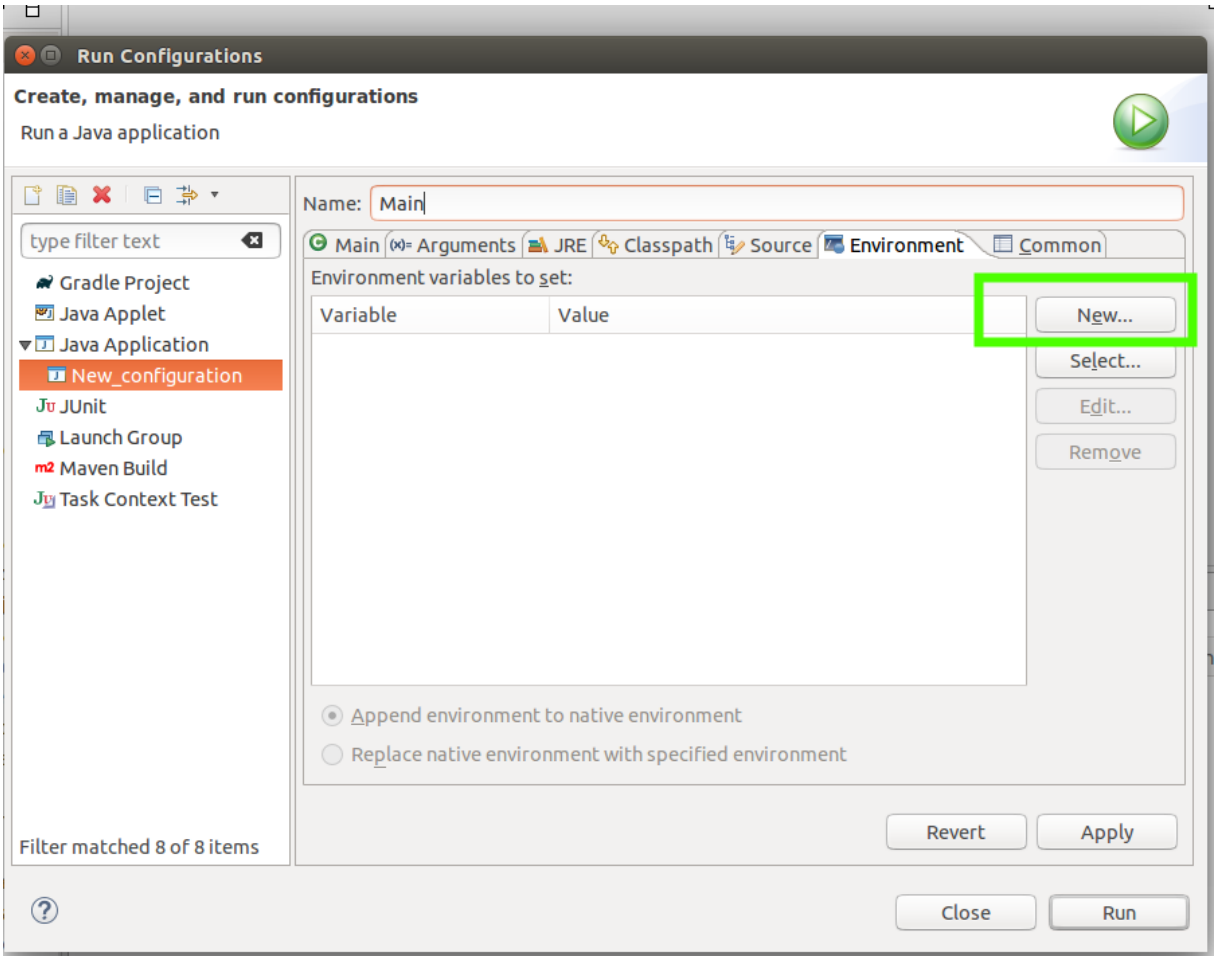
- Once you add it it should look like this



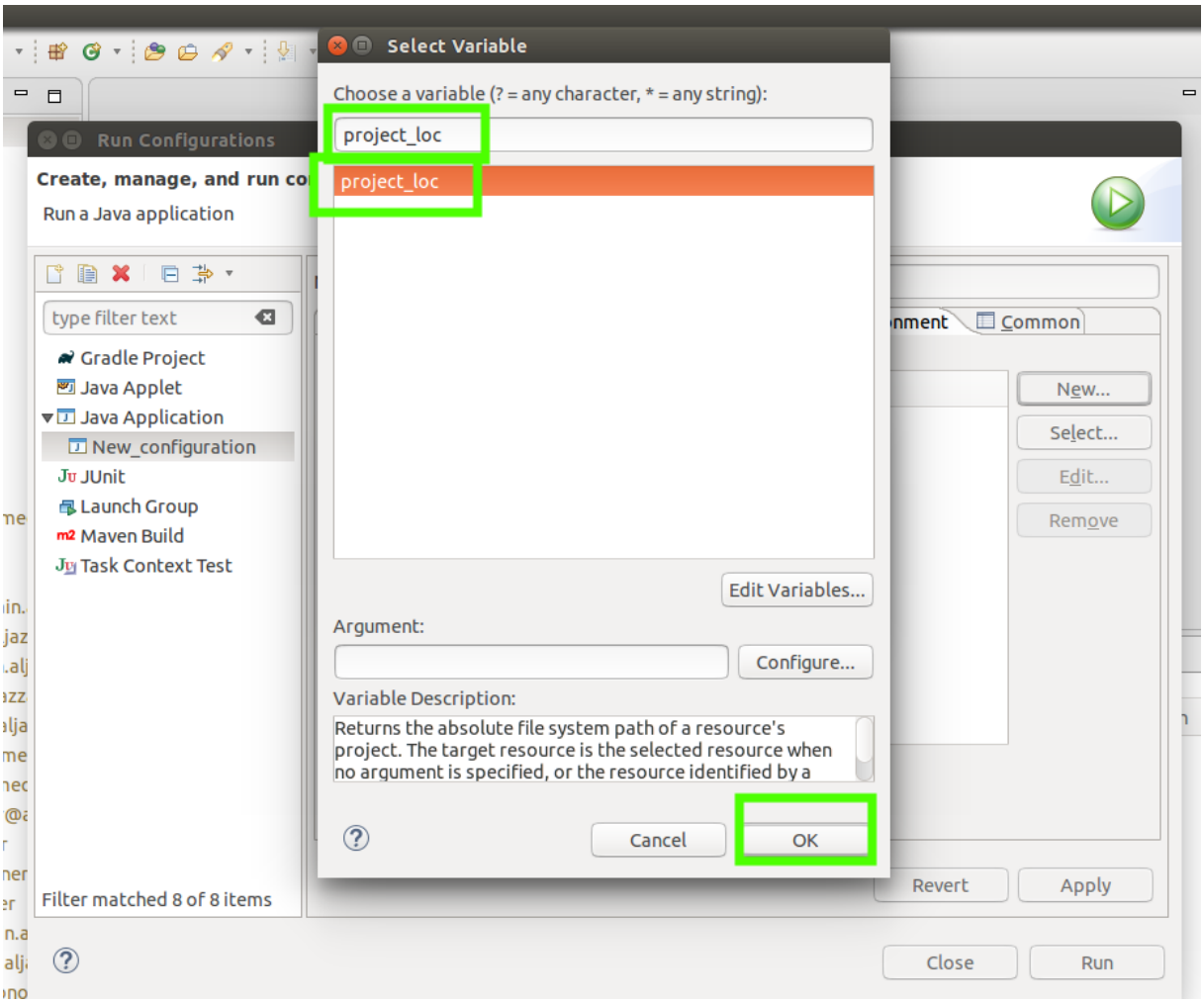
- Now click on arguments and as for arguments write : **-prism embedded.sm embedded.csl**



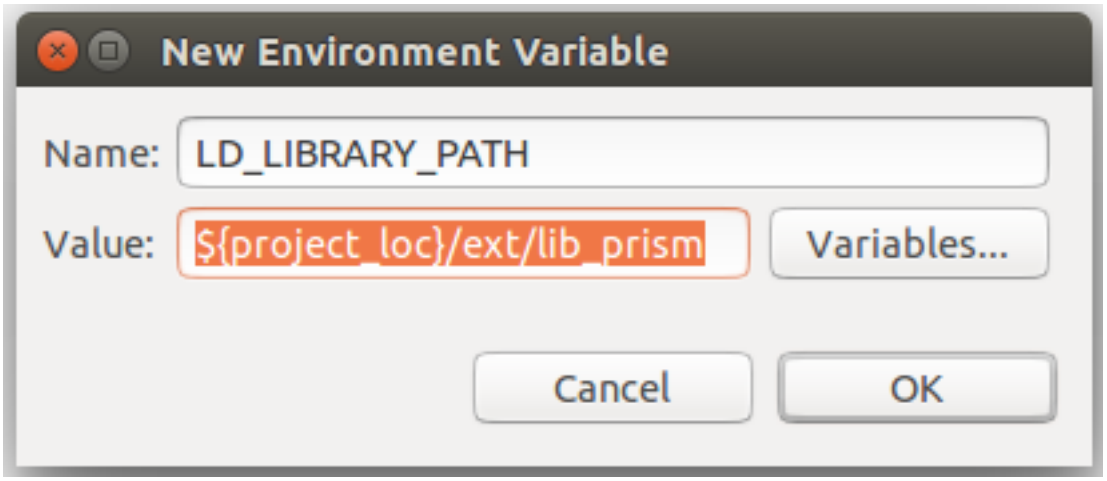
- You dont have to change **JRE,ClassPath,Source and Common** sections, open the **Environment** section



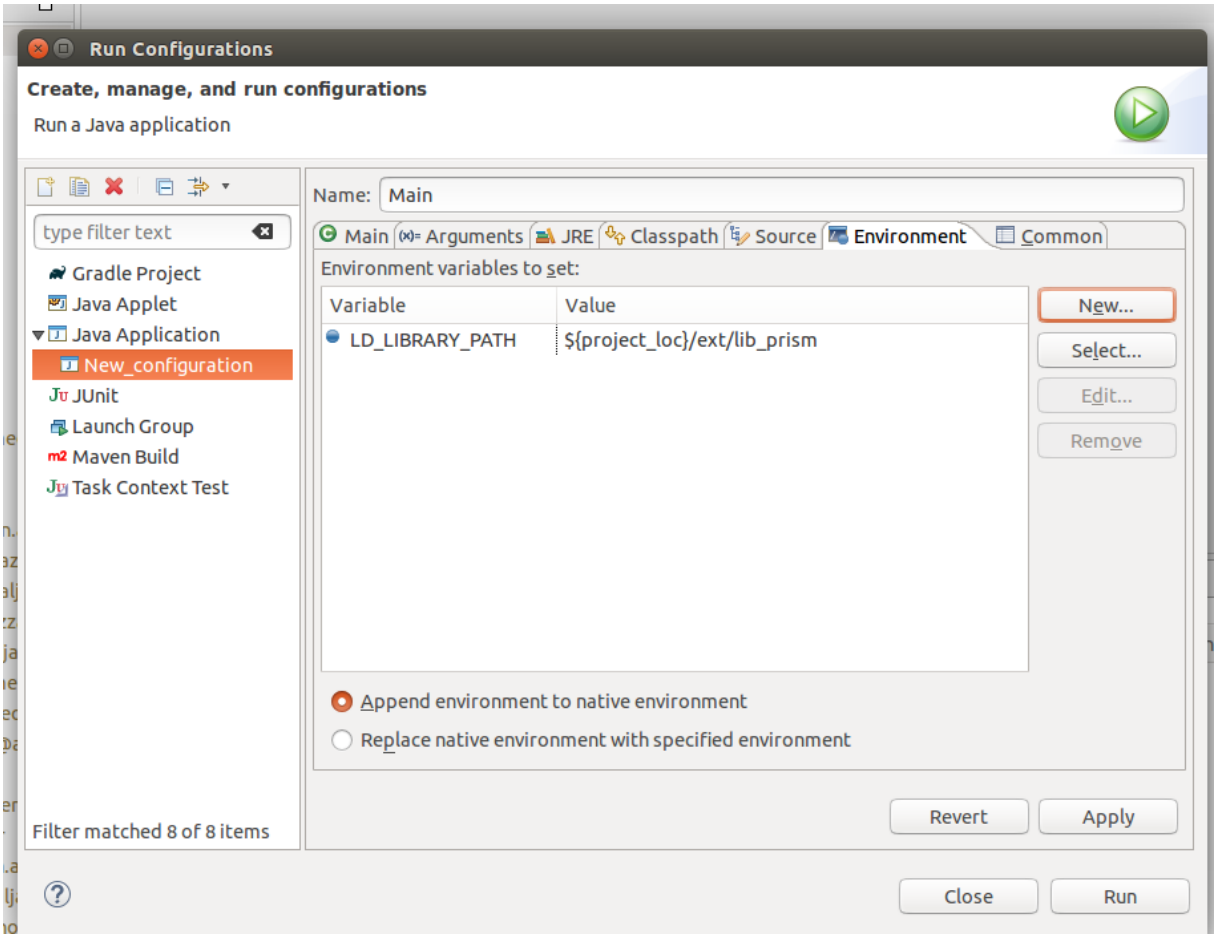
- Click on **New** –> Name should be **LD_LIBRARY_PATH** and click on **variables** and then type **project_loc**, add it.



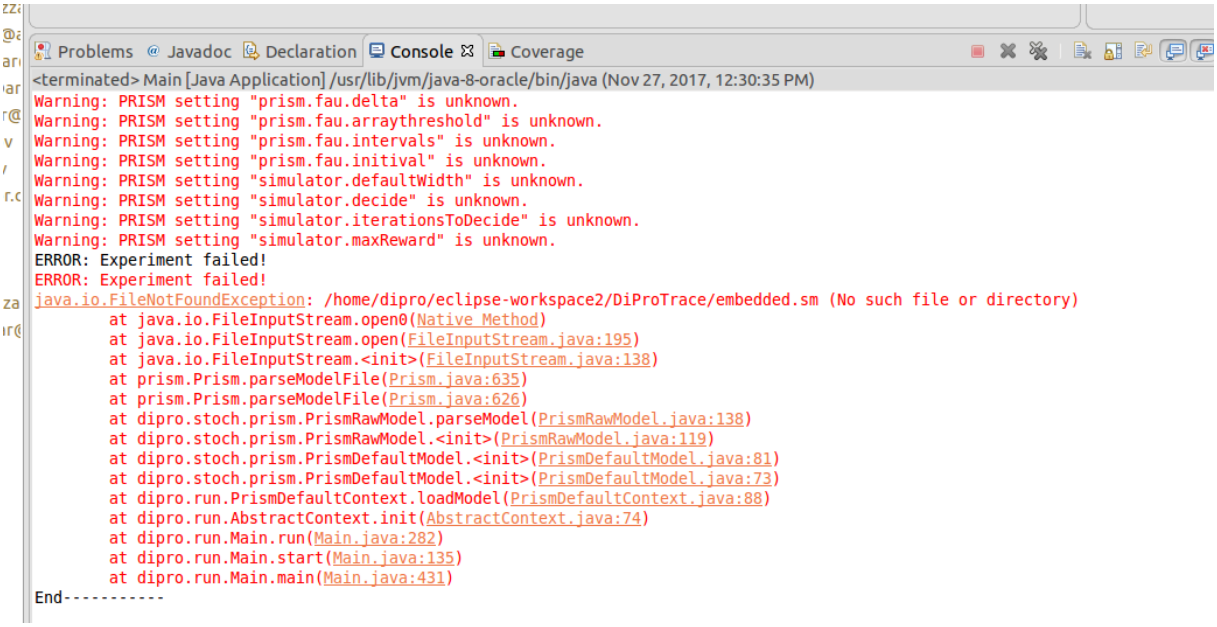
- Modify variable name as **${project_loc}/ext/lib_prism**



- It should be in such a form.

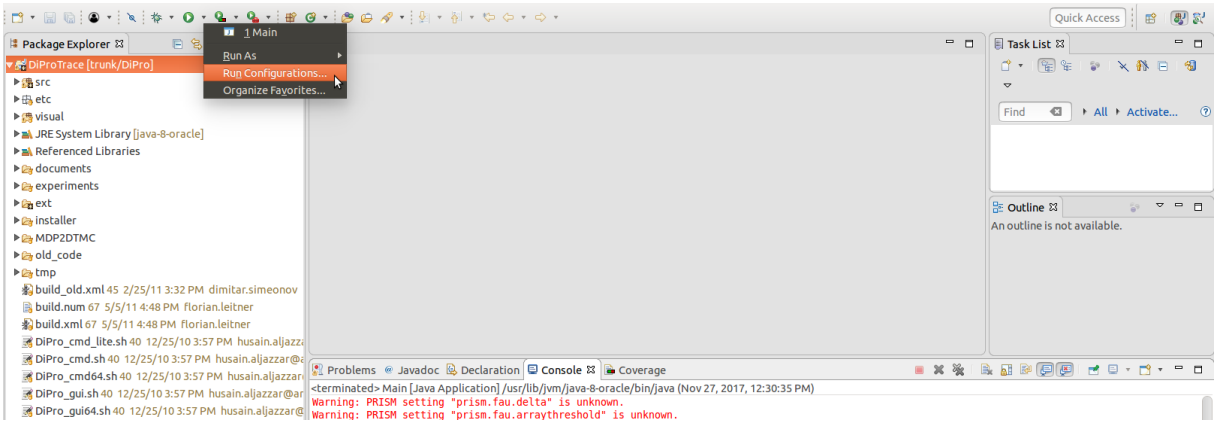


- Now click on **Apply** and then **Run**

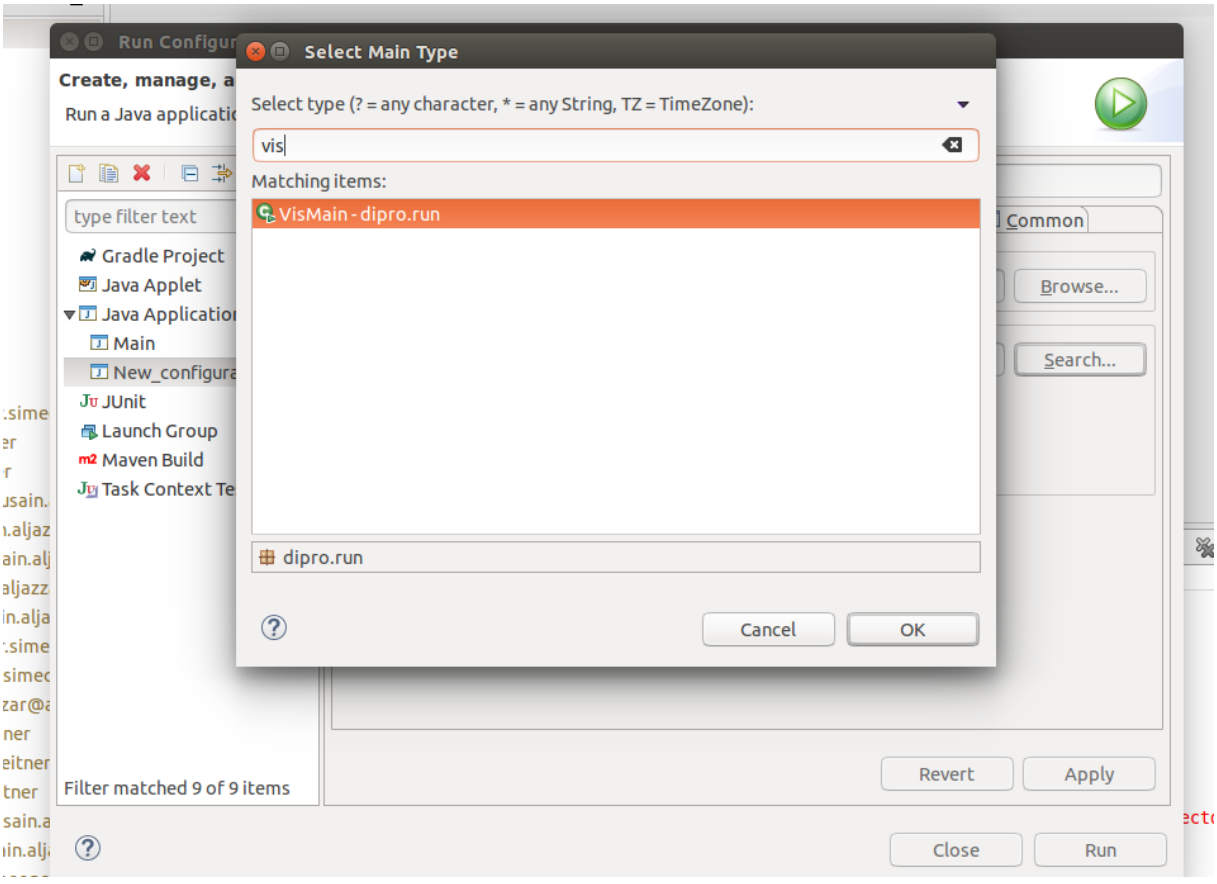


- When you run the program if you getting above error it means everything is fine because as you can see it just simply says that you have no embedded.sm file for the experiment you should add it manually to project folder.
- Now we can create the visual GUI for the Dipro

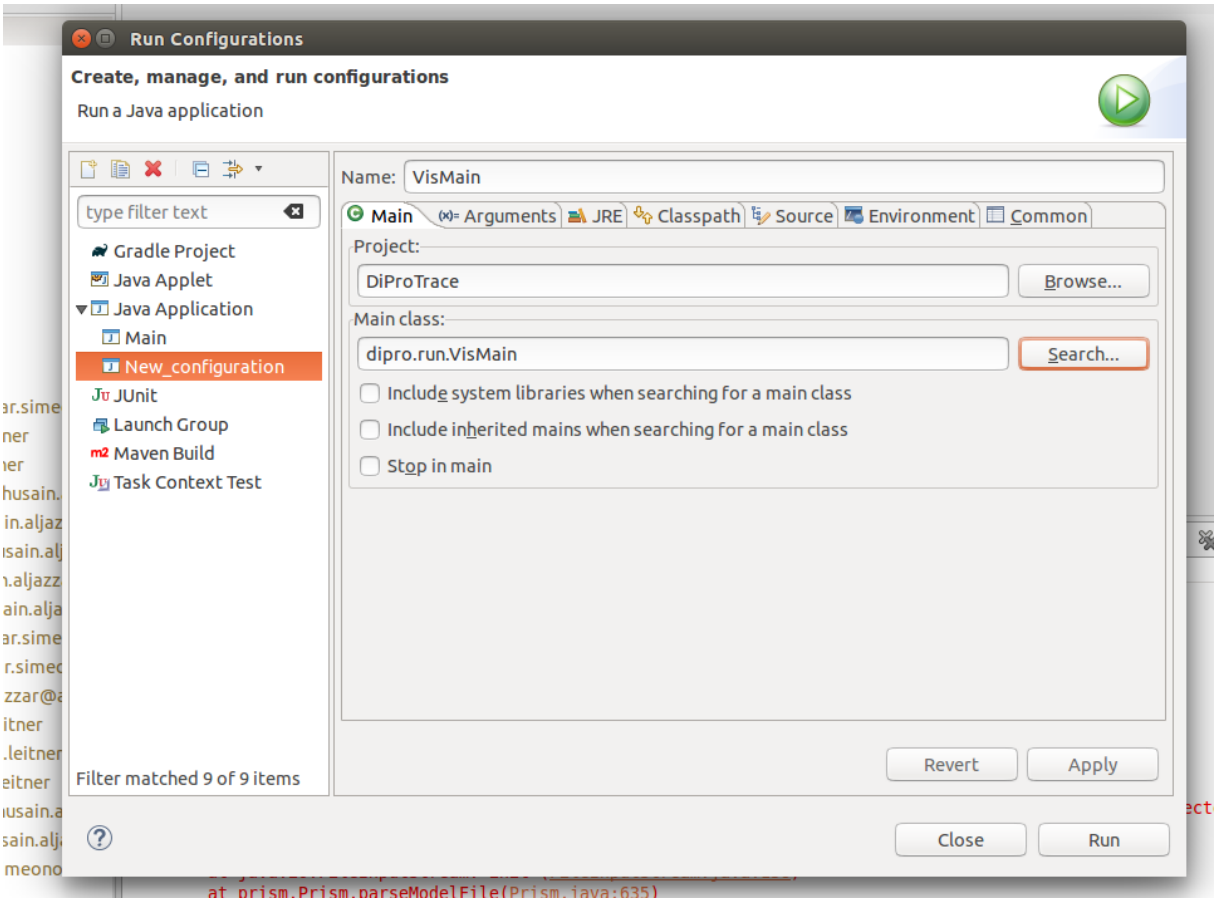
- Click on the small little triangular next to run as and select **Run Configurations.."
.



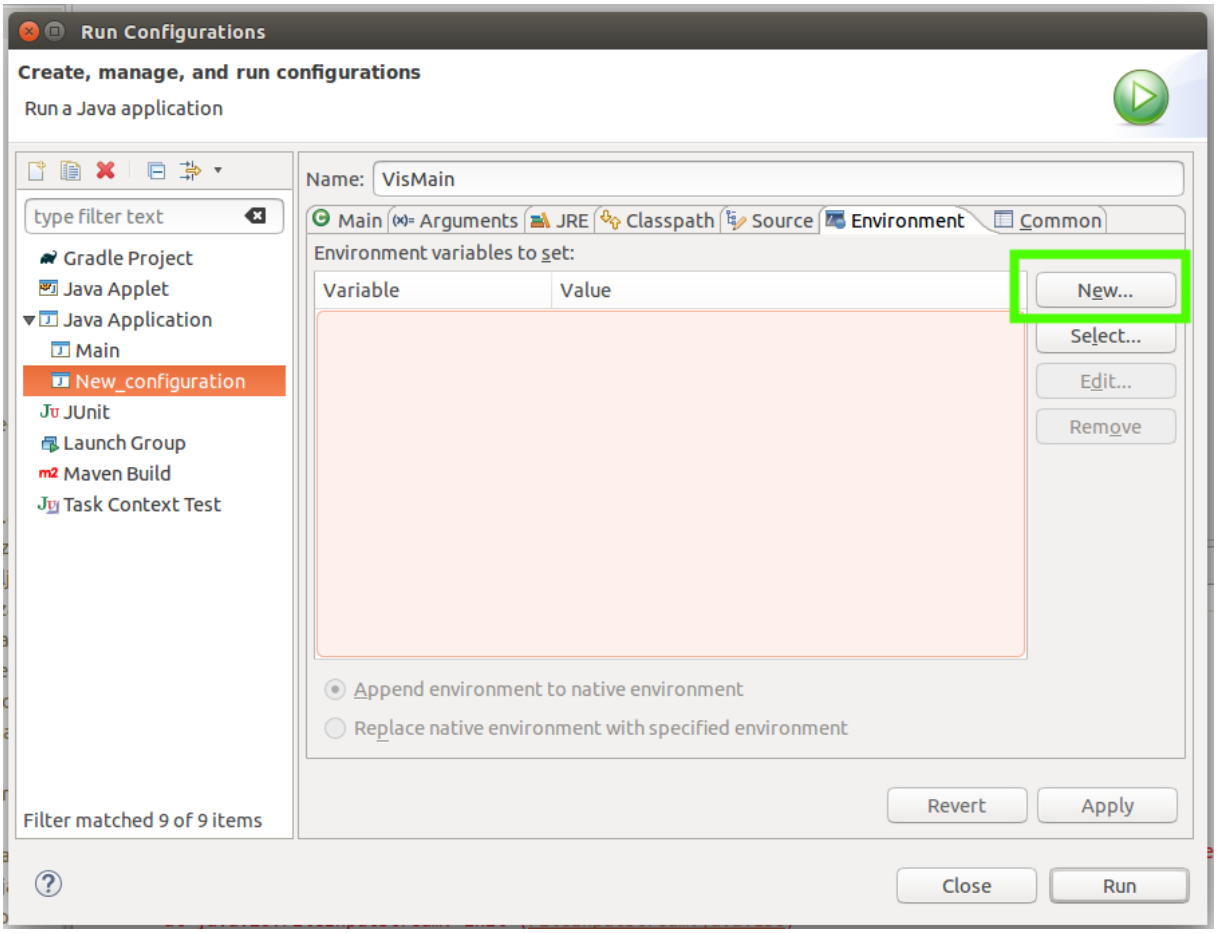
- Double Click on Java Application again.
- Name it as **VisMain** as for Project select the **DiProTrace** and for main class click on search and type **vis** and add it.



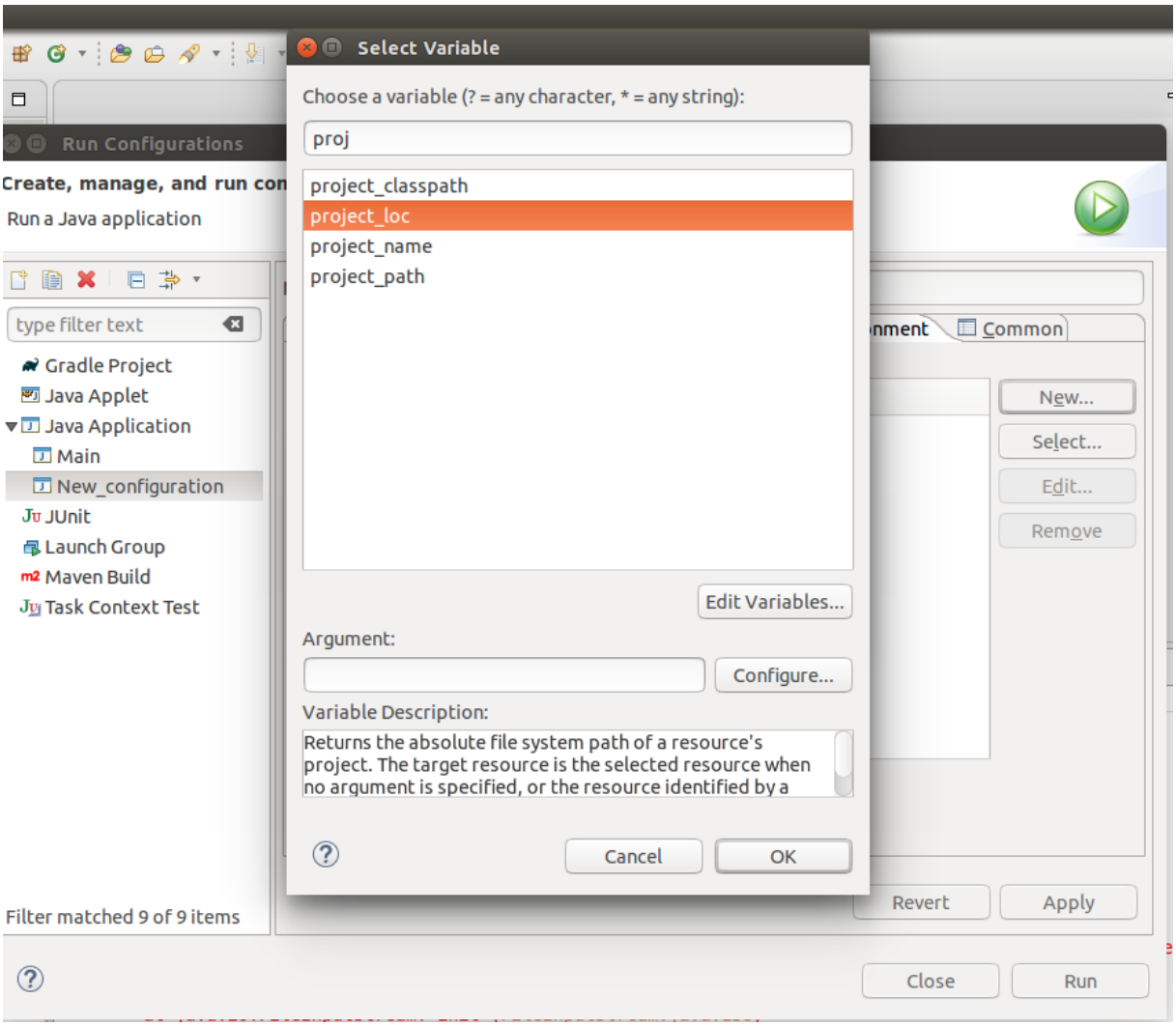
- General view would be as following image.



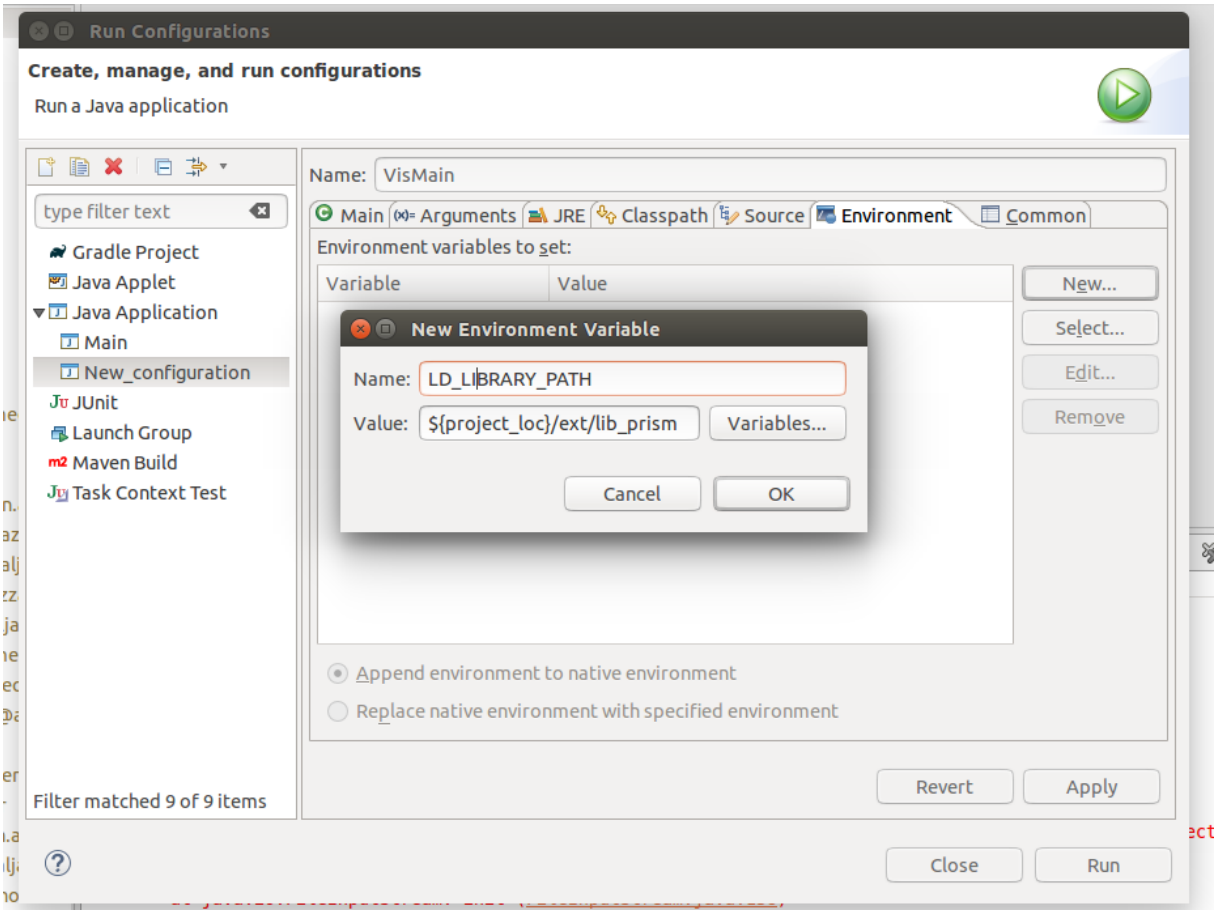
- Change to Environment section and click on **New**



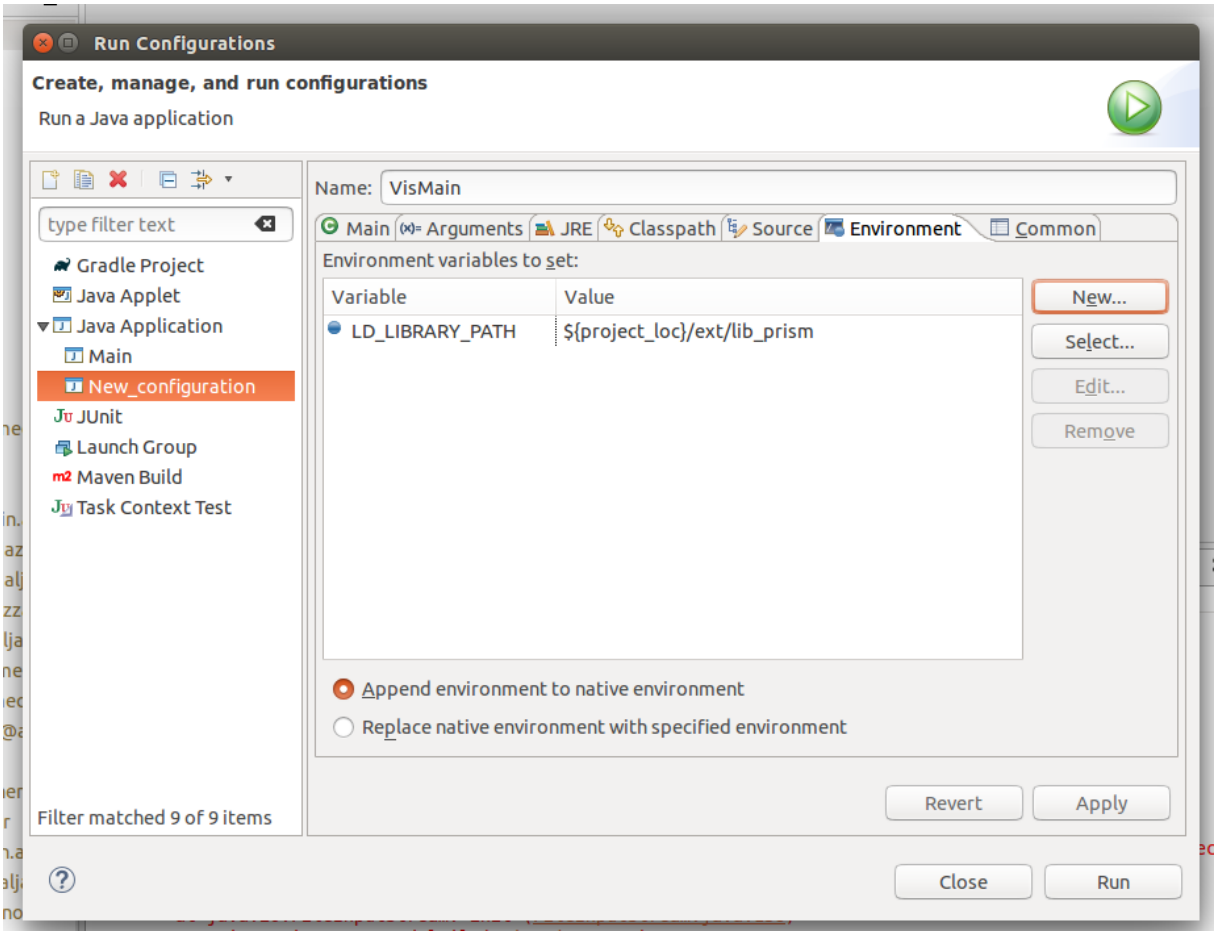
- As we did before, name should be **LD_LIBRARY_PATH** and for value click on **variables** write project_loc and select that.



- Modify variable name as **${project_loc}/ext/lib_prism**



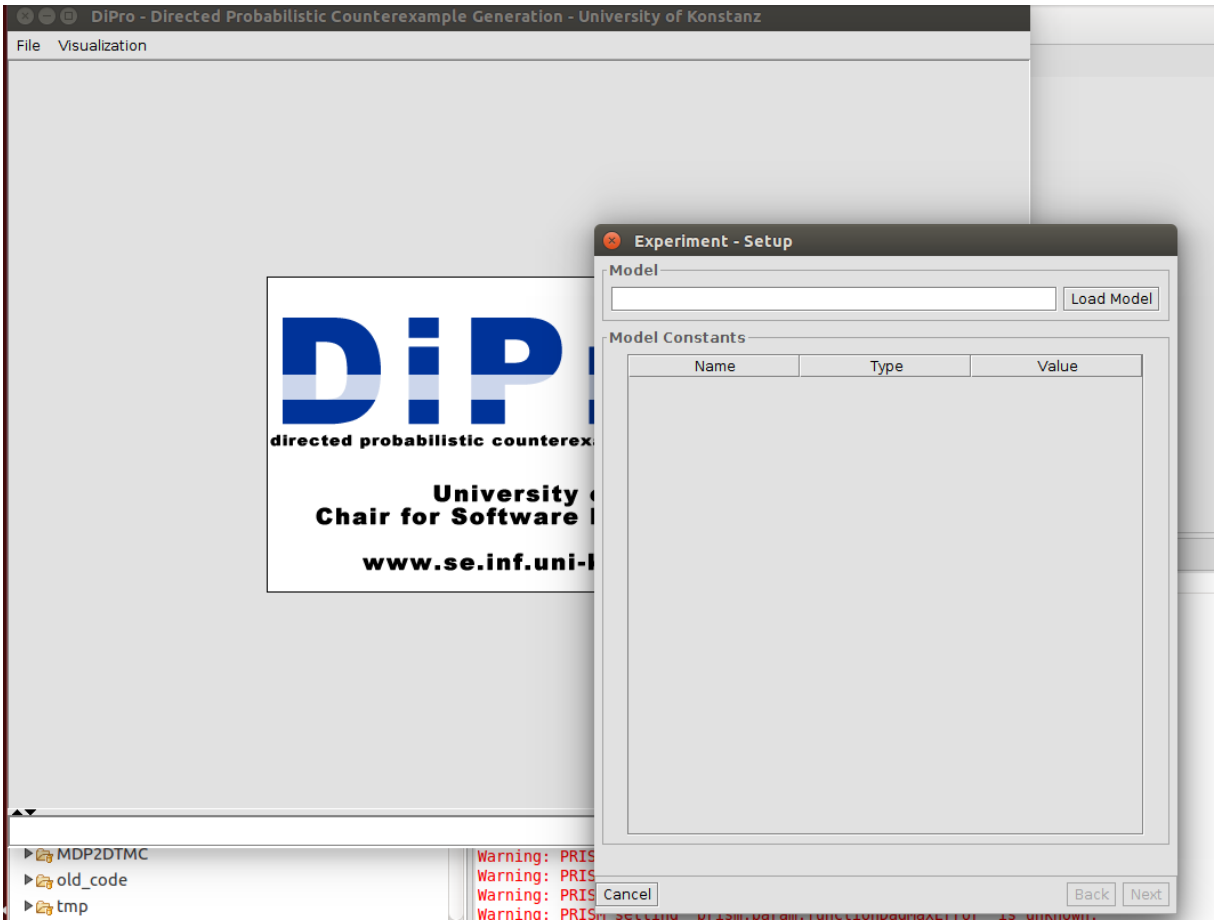
- At the end it should be as following:



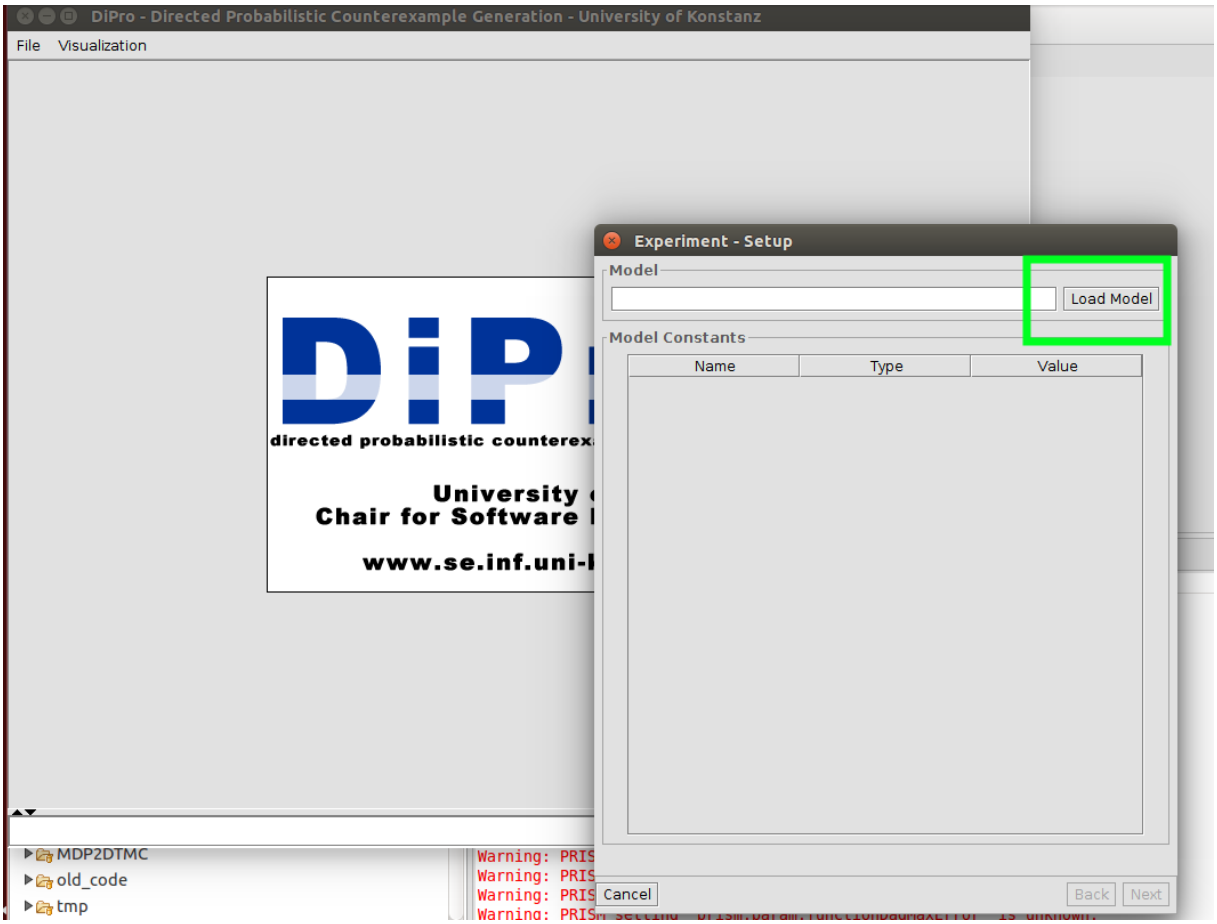
- Apply it and click on Run

7 Demonstration of Running DiPro

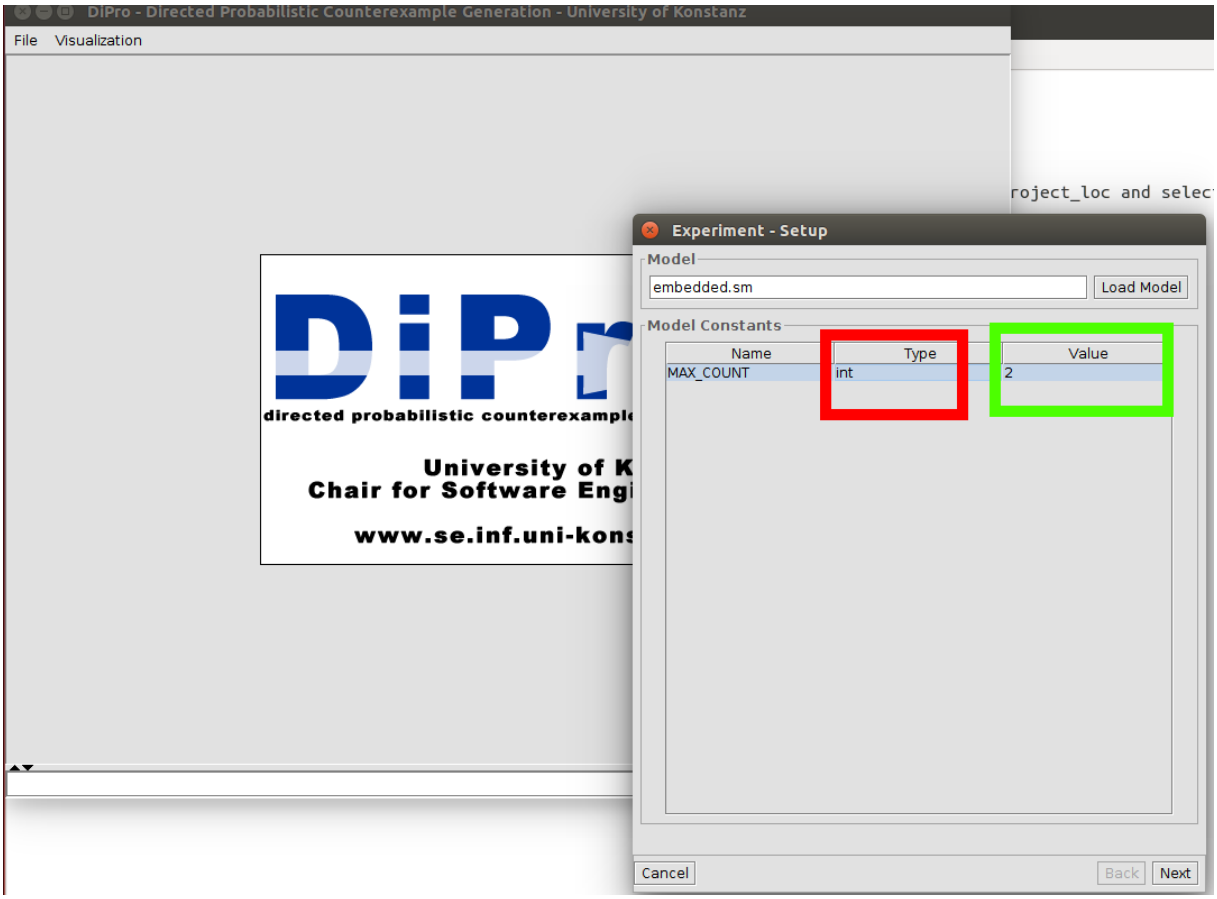
Once you run the program you'll be welcomed by the program.



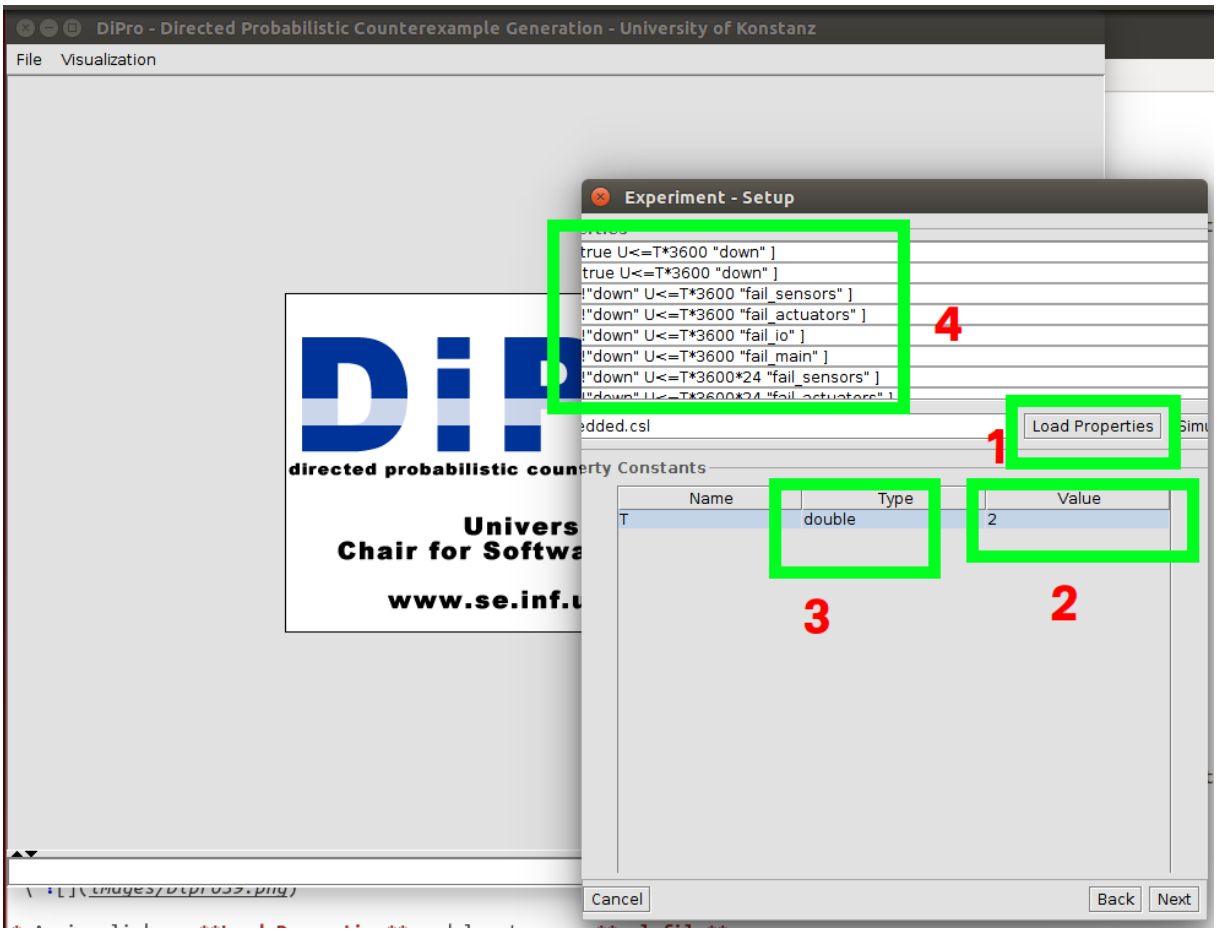
- Click on **Load Model**



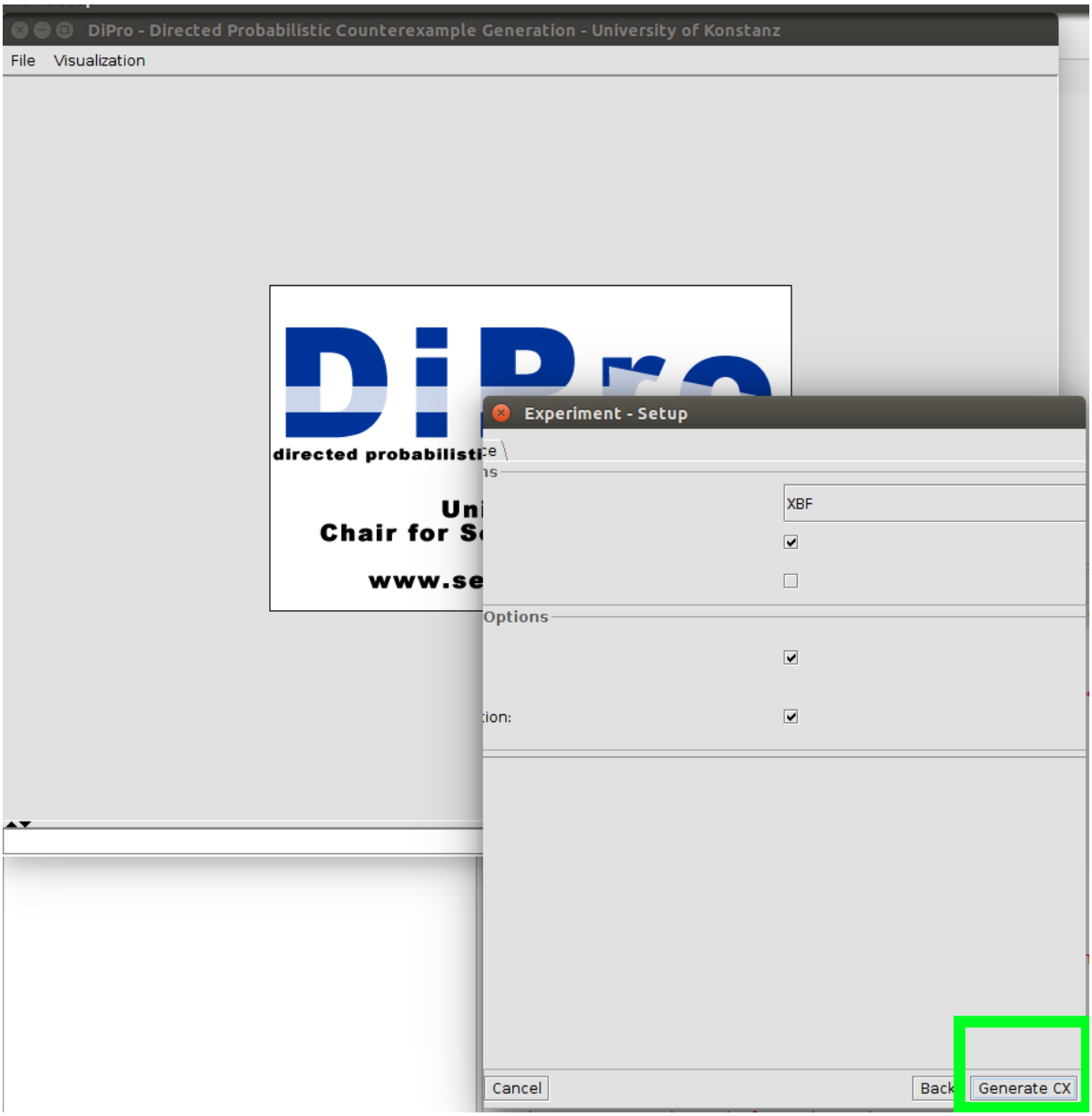
- Now you should load your .sm file , in our case it is located at the default workspace so we navigate to it.
- Once we load the model, click on the row under the **value column** and enter value like 2, and then click under the column of **Type**.Then **Next** button should be activated.



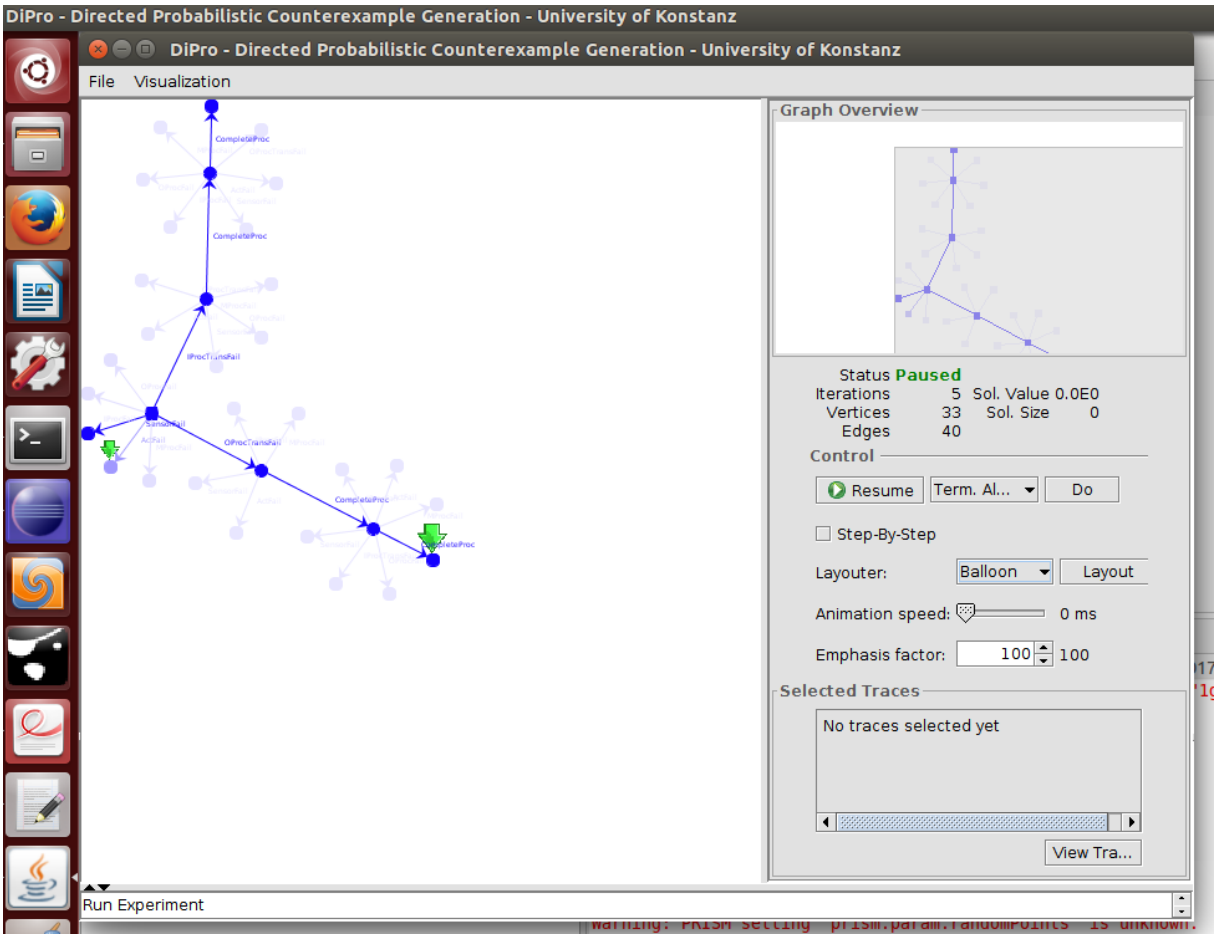
- Again click on **Load Properties** and locate your **csl file** as first step. Secondly, enter a value as shown in the picture below, then click on Type variable such as double as shown in the picture. and lastly click a specific property and then next button should be activated again. Click on next to proceed.



- Now leave it default and continue with the **Generate CX**



- As you can see program is running now.

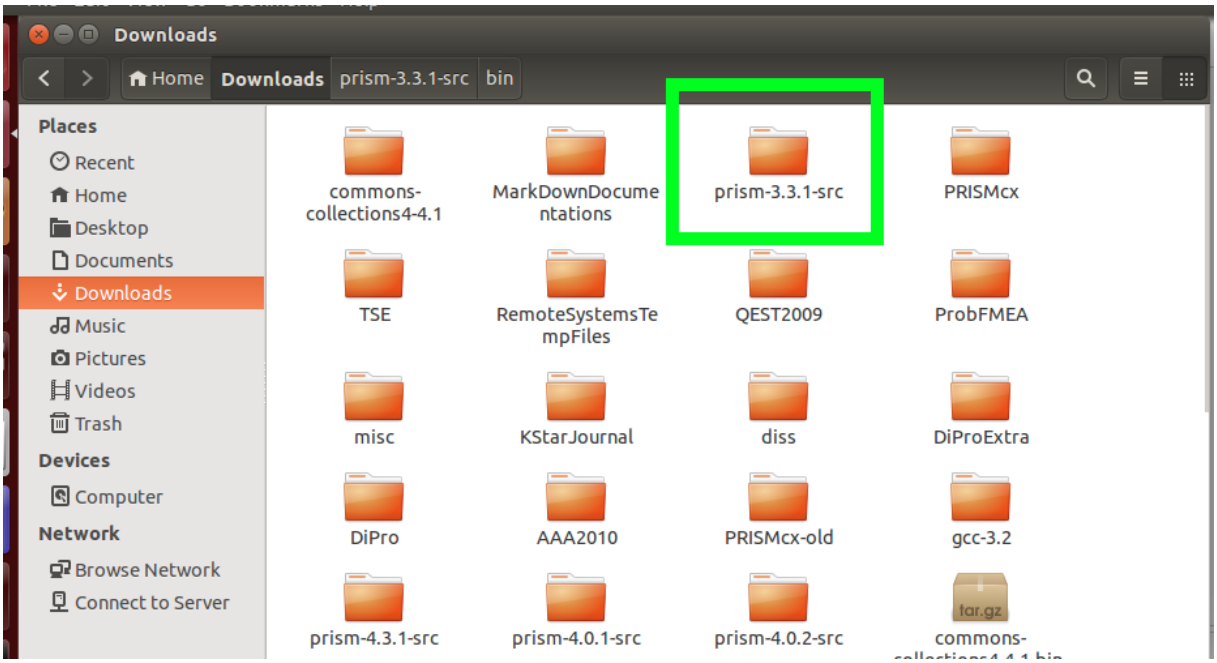


8 FAQ(Frequently Asked Questions)

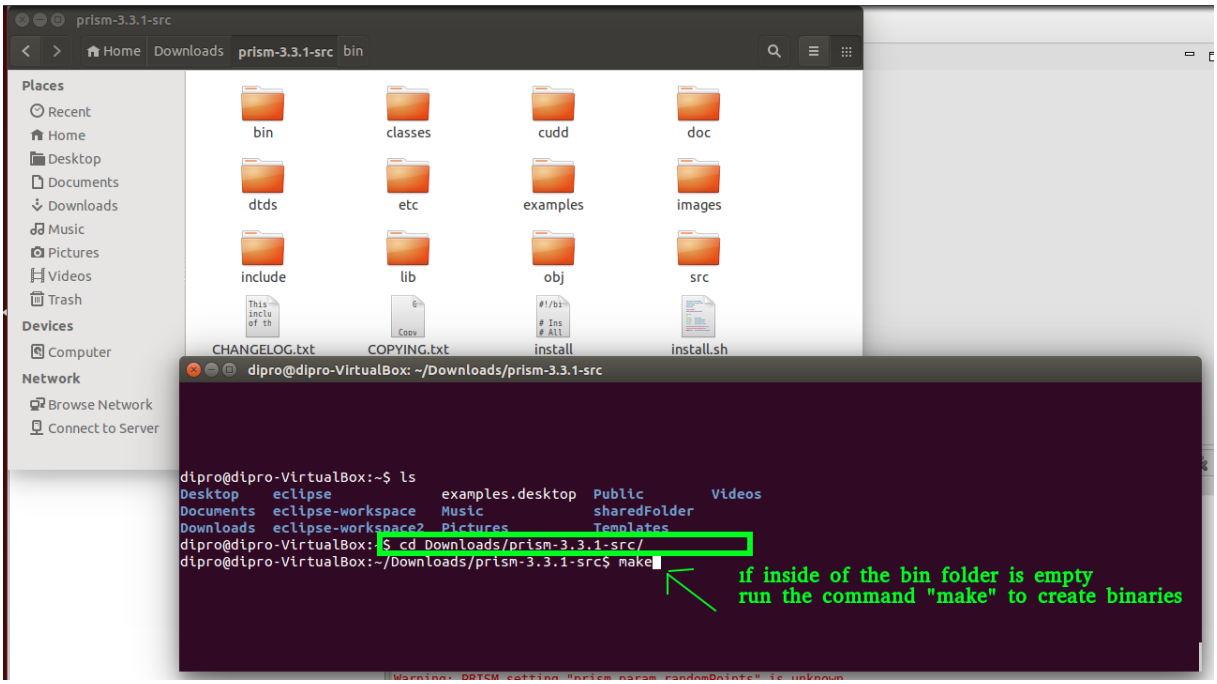
8.1 How can I use different version of the prism with Dipro?

In order to use different version of prism with Dipro we should make small changes in the project or else it is higly natural that we will get dependency errors based on library differences. Before getting into that, you should make sure to generate prism.jar file in the prism folder's lib directory.In order to avoid possible problems please make sure to follow these steps:

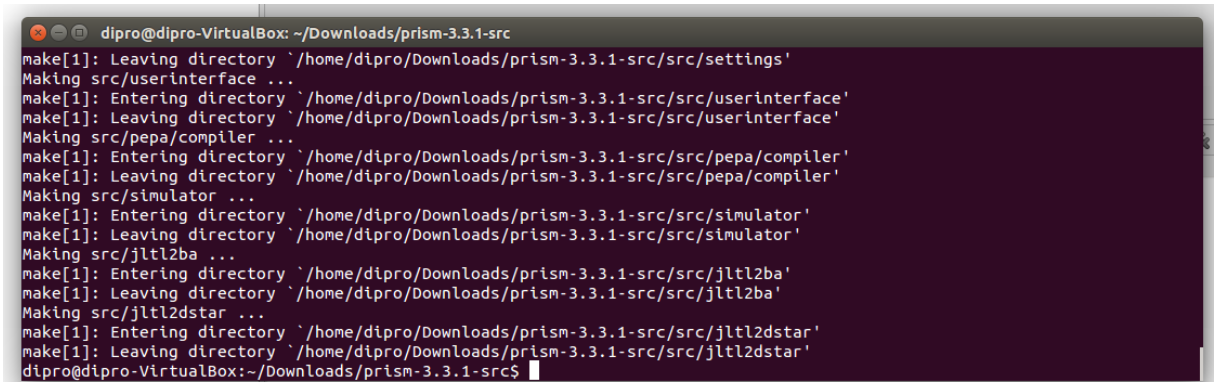
- Locate the prism folder you want to use and navigate inside the folder.



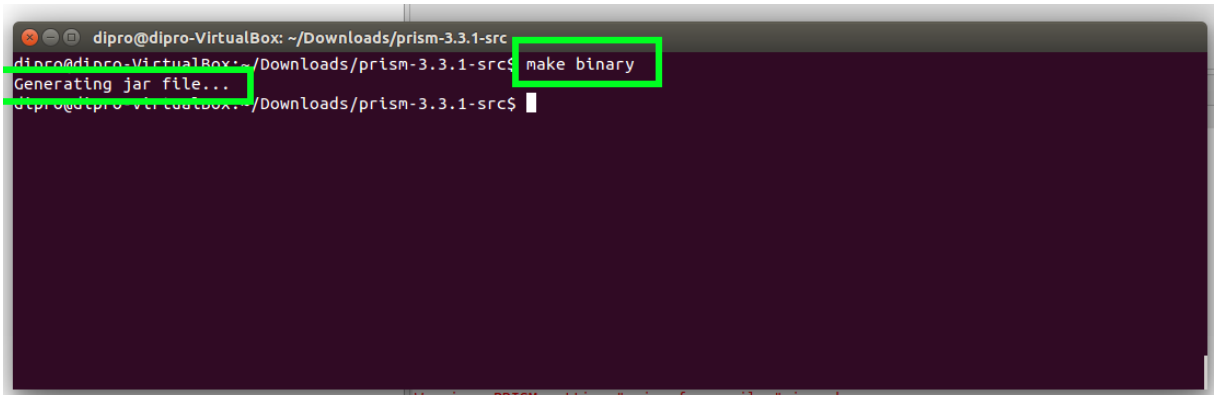
- Look inside to **bin** folder. If there are no files in that, open the terminal, enter the prism file with cd commands by navigating and run the command **make**



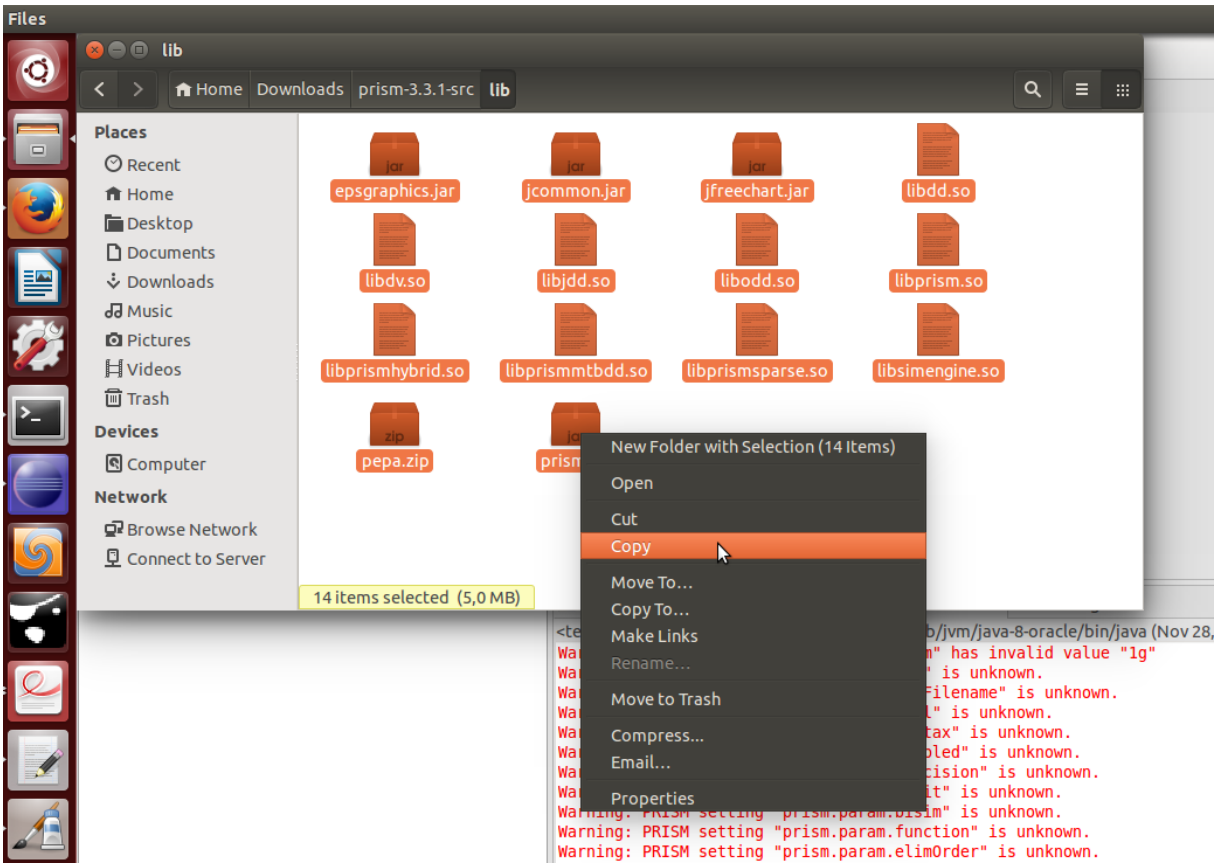
- It'll generate the binaries for you.



- Now check the content of lib folder, navigate inside the folder and control if there is a prism.jar exists. If not we have to generate it. In order to generate prism.jar file again open the project location inside the terminal and run command make binary and then you can make sure that **prism.jar** is generated inside the lib folder.



- Now once you complete all of these above steps, open your lib directory inside the prism folder and then copy everything.



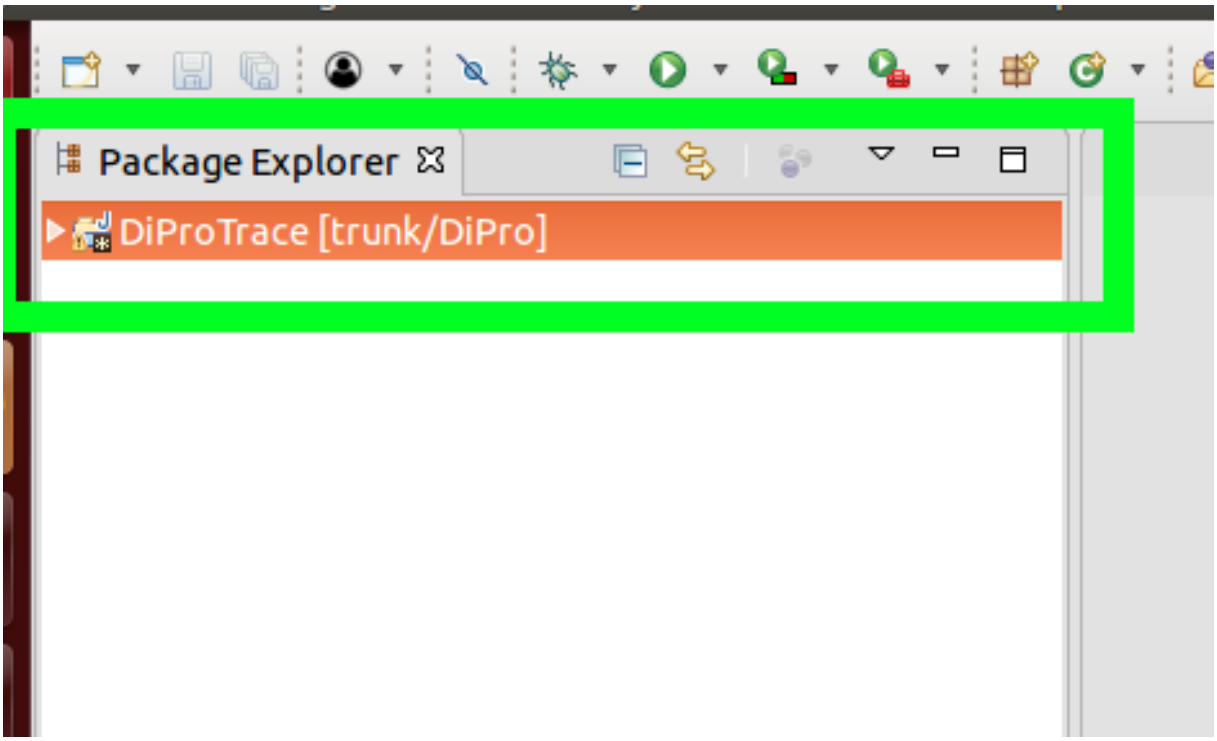
- Now go to your eclipse workspace and project folder, open up the project folder navigate to /workspace/DiProTrace/ext/lib_prism
- Here change everything by pasting the copied content of prism lib folder.
- As you finish your replacement, you can restart the eclipse ide and refresh the project by right clicking on it in the project explorer and hitting **refresh**
- If you still get any errors.Right click on the project ,navigate to Java Build Path –> Libraries and add them manually by clicking to **Add Jars..** and selecting them from prism_lib folder.

8.2 Java.lang.UnsatisfiedLinkError: no prism in java.library.path

This error occurs because you need to set your java library path with the corresponding prism directory where your actual prism version is located. As we already have our prism.jar file inside the **lib_prism** folder ,we need to state it explicitly. For that , we can use **${project_loc}/ext/lib_prism** value and variable name should be LD_LIBRARY_PATH for linux users and LYLD_LIBRARY_PATH for macOS users. Here what you need to keep in mind that, no matter where your prism.jar file is located, you should give the path as value for it. Therefore it should be something like /path/to/your/jar/file.

8.3 "Launching VisMain or Main" has encountered a problem. Variable references empty selection : ${project_loc}

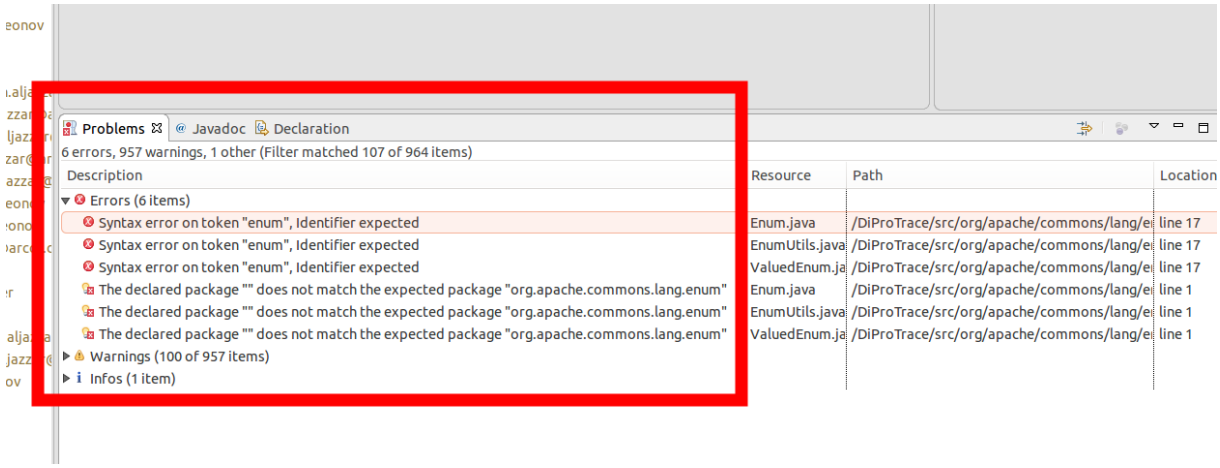
This specific error usually happens whenever our project folder on the project explorer is deselected for somereason or simply when we open the Eclipse ide for the very first time again project folder is deselected again. Afterward, when you want to run the program it wants to use project_loc variable and since project is deselected it does not explicitly know how to load this variable. To fix this problem, before running the DiPro source code please make sure to select project folder on the project explorer. Once you select it,it should be selected and kinda highlighted by orange light before we run it.



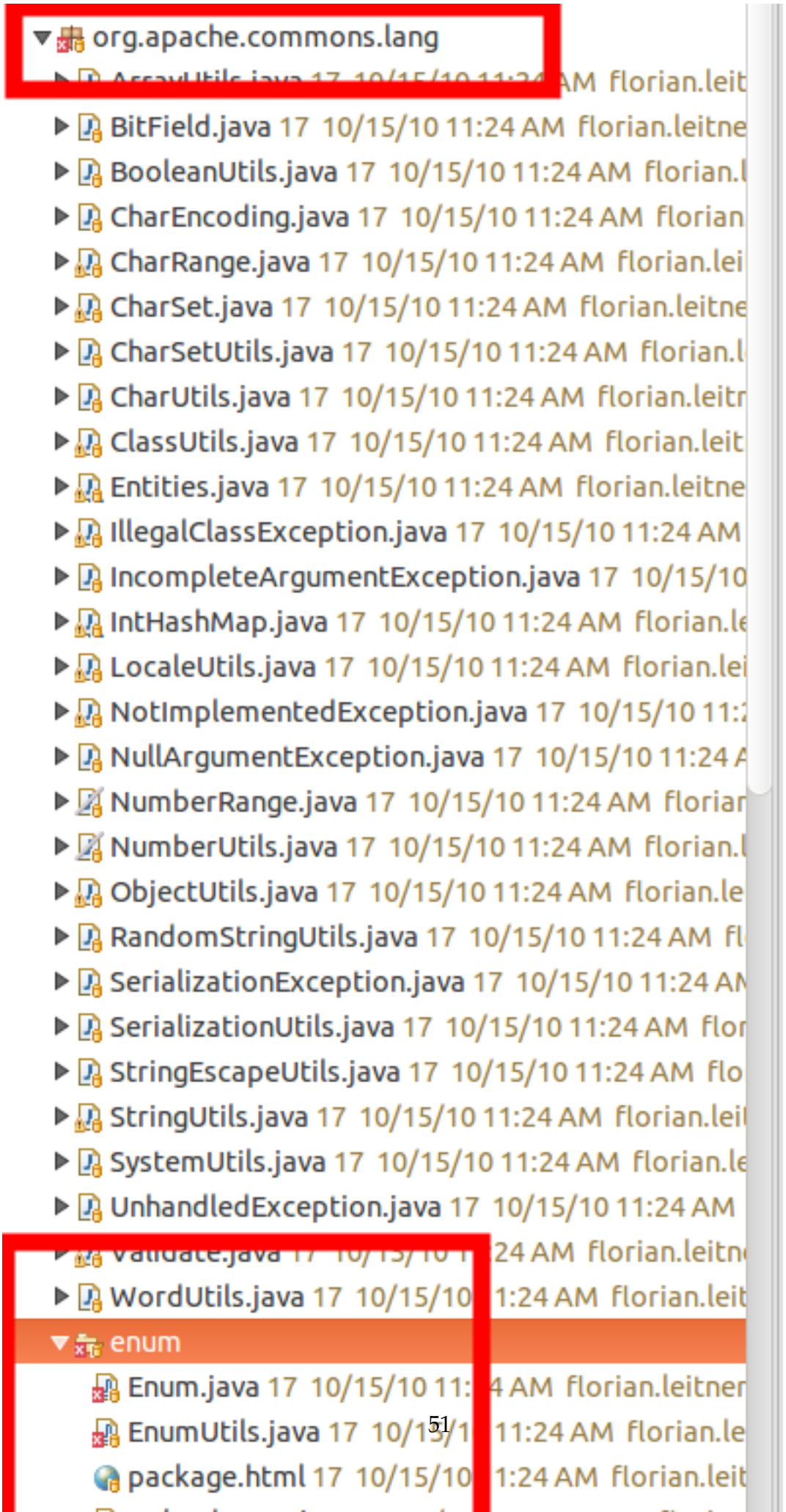
8.4 The Project was not built since its build path is incomplete.Cannot find the class file for x.y.z.t

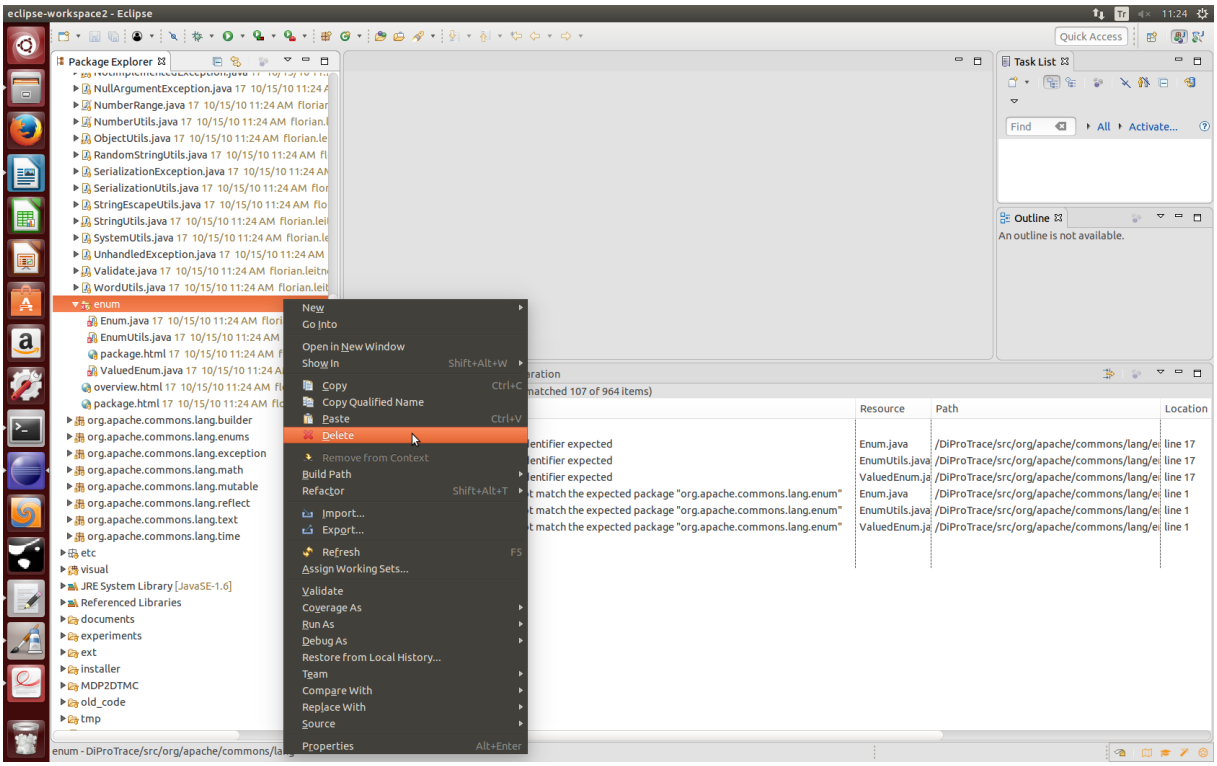
This is a very common error when changing libraries from one prism version to another. As the error says, it simply says all meaning that there is a missing library in your class path, first focuse on the second part of the error where it is state which class is missing. Dig into it, lets say x.y.z.t is missing here you can track that down to bottom and see what's exactly missing for project and that causes to compilation error. Try downloading the jar file causing to it. Once you download it right click on the project folder inside the Eclipse Project explorer view, and click on Properties. In the new window, click on Java Build Path and come to libraries. Here you should manually add the jar file by clickin on Add jar(if you put the jar file inside the project) or add external jar(if jar file is located for example in ur downloads folder). Once you add the required jar, let it rebuild and then it'll be error free.

8.5 Eclipse Java syntax error on token “enum”,identifier expected and The declared package “” does not match the expectedpackage “org.apache.commons.lang.enum” errors

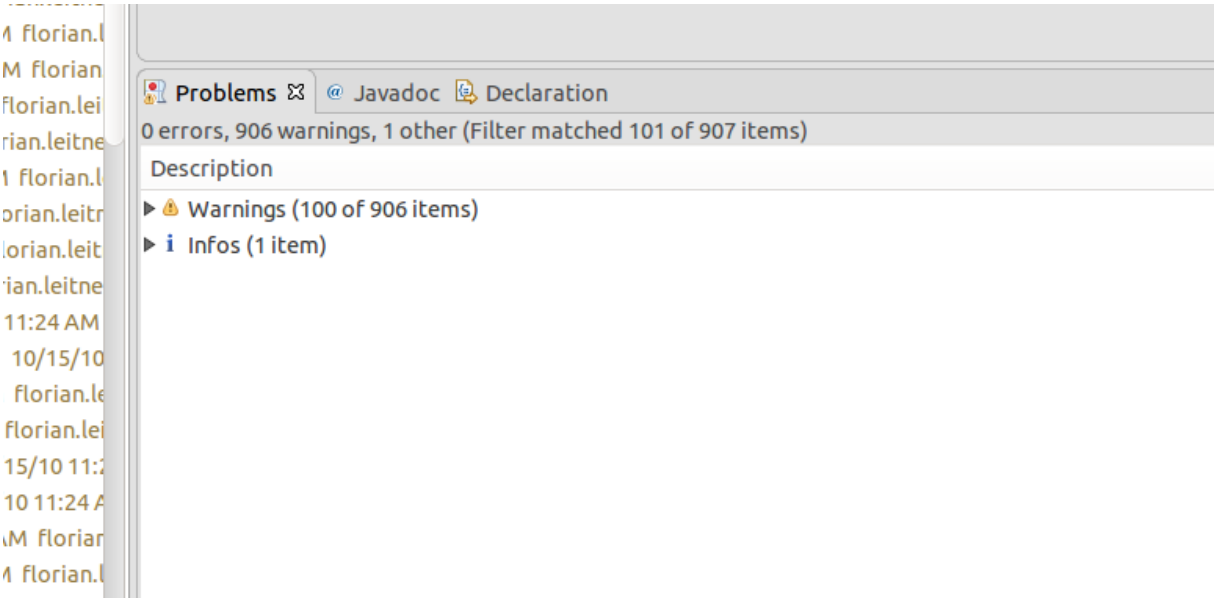


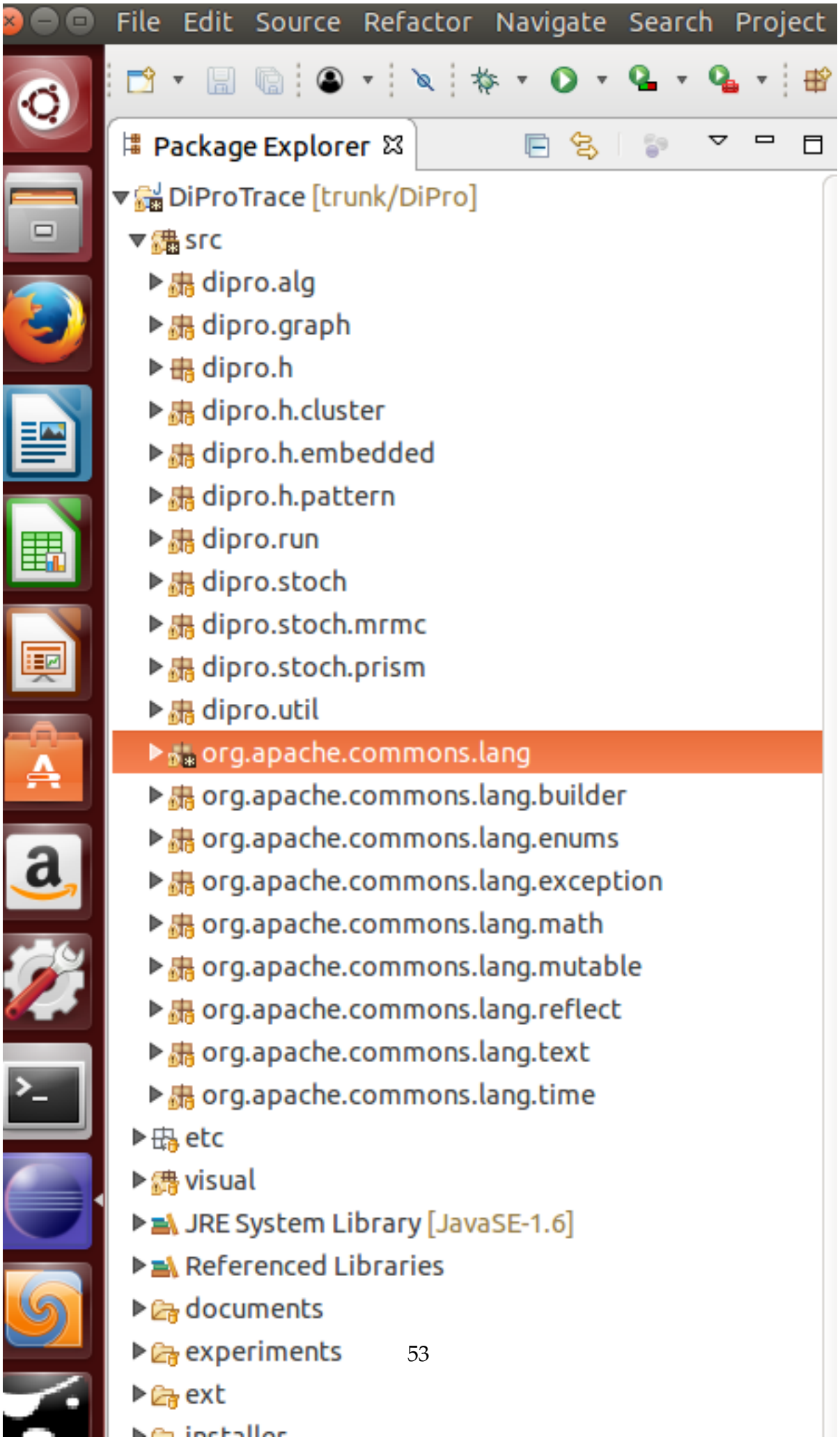
- We get this error because the usage of enum keyword is depricated by java 5 and replaced with the current one called enums.This project includes both **enum(depricated)** and **enums(updated one)** packages.Therefore easiest way to fix this problem is removing enum files from the project as we already have the current one called enums.
- To remove depricated enum in our system please follow the upcoming steps respectively.
- If you take a look at project explorer on the left hand side, there is still a red cross sign indicating the error causing this, therefore we should remove this folder causing to the error.
- First navigate to project folder and expand on the error causing to this. You can do this by simply expanding, **DiProTrace[trunk/DiPro]** –> **src** –> **org.apache.commons.lang** –> **enum**
- Now all we have to do is right click on this folder and delete it.


org.apache.commons.lang
BitField.java 17 10/15/10 11:24 AM florian.leitne
BooleanUtils.java 17 10/15/10 11:24 AM florian.l
CharEncoding.java 17 10/15/10 11:24 AM florian
CharRange.java 17 10/15/10 11:24 AM florian.lei
CharSet.java 17 10/15/10 11:24 AM florian.leitne
CharSetUtils.java 17 10/15/10 11:24 AM florian.l
CharUtils.java 17 10/15/10 11:24 AM florian.leitr
ClassUtils.java 17 10/15/10 11:24 AM florian.leit
Entities.java 17 10/15/10 11:24 AM florian.leitne
IllegalClassException.java 17 10/15/10 11:24 AM
IncompleteArgumentException.java 17 10/15/10
IntHashMap.java 17 10/15/10 11:24 AM florian.le
LocaleUtils.java 17 10/15/10 11:24 AM florian.lei
NotImplementedException.java 17 10/15/10 11:
NullArgumentException.java 17 10/15/10 11:24 A
NumberRange.java 17 10/15/10 11:24 AM florian
NumberUtils.java 17 10/15/10 11:24 AM florian.l
ObjectUtils.java 17 10/15/10 11:24 AM florian.le
RandomStringUtils.java 17 10/15/10 11:24 AM fl
SerializationException.java 17 10/15/10 11:24 AM
SerializationUtils.java 17 10/15/10 11:24 AM flor
StringEscapeUtils.java 17 10/15/10 11:24 AM flo
StringUtils.java 17 10/15/10 11:24 AM florian.leit
SystemUtils.java 17 10/15/10 11:24 AM florian.le
UnhandledException.java 17 10/15/10 11:24 AM
Validate.java 17 10/15/10 11:24 AM florian.leitne
WordUtils.java 17 10/15/10 11:24 AM florian.leit
enum
Enum.java 17 10/15/10 11:24 AM florian.leitner
EnumUtils.java 17 10/15/10 11:24 AM florian.le
package.html 17 10/15/10 11:24 AM florian.leit



- When you click on okey and delete the depricated enum folder, notice that our project is now free of errors and also on the left hand side there are no more red cross on the project explorer view.




File Edit Source Refactor Navigate Search Project
Package Explorer
DiProTrace [trunk/DiPro]
src
dipro.alg
dipro.graph
dipro.h
dipro.h.cluster
dipro.h.embedded
dipro.h.pattern
dipro.run
dipro.stoch
dipro.stoch.mrmc
dipro.stoch.prism
dipro.util
org.apache.commons.lang
org.apache.commons.lang.builder
org.apache.commons.lang.enums
org.apache.commons.lang.exception
org.apache.commons.lang.math
org.apache.commons.lang.mutable
org.apache.commons.lang.reflect
org.apache.commons.lang.text
org.apache.commons.lang.time
etc
visual
JRE System Library [JavaSE-1.6]
Referenced Libraries
documents
experiments
ext