

How to Develop Geronimo Eclipse Plugin in Eclipse

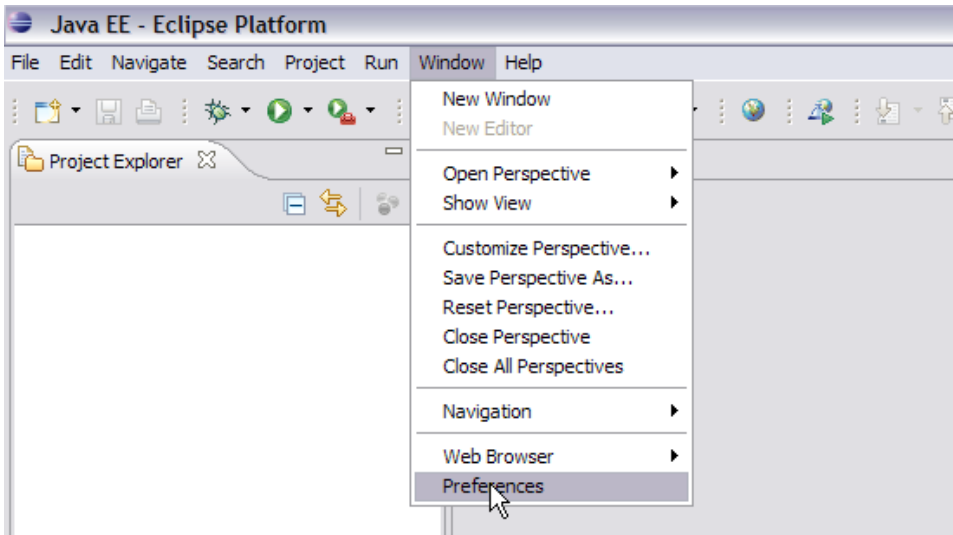
[How to Build Geronimo Eclipse Plugin from Source](#)

[Apache Geronimo Development Tools Project](#) [How to get daily Geronimo Eclipse Plugin builds](#)

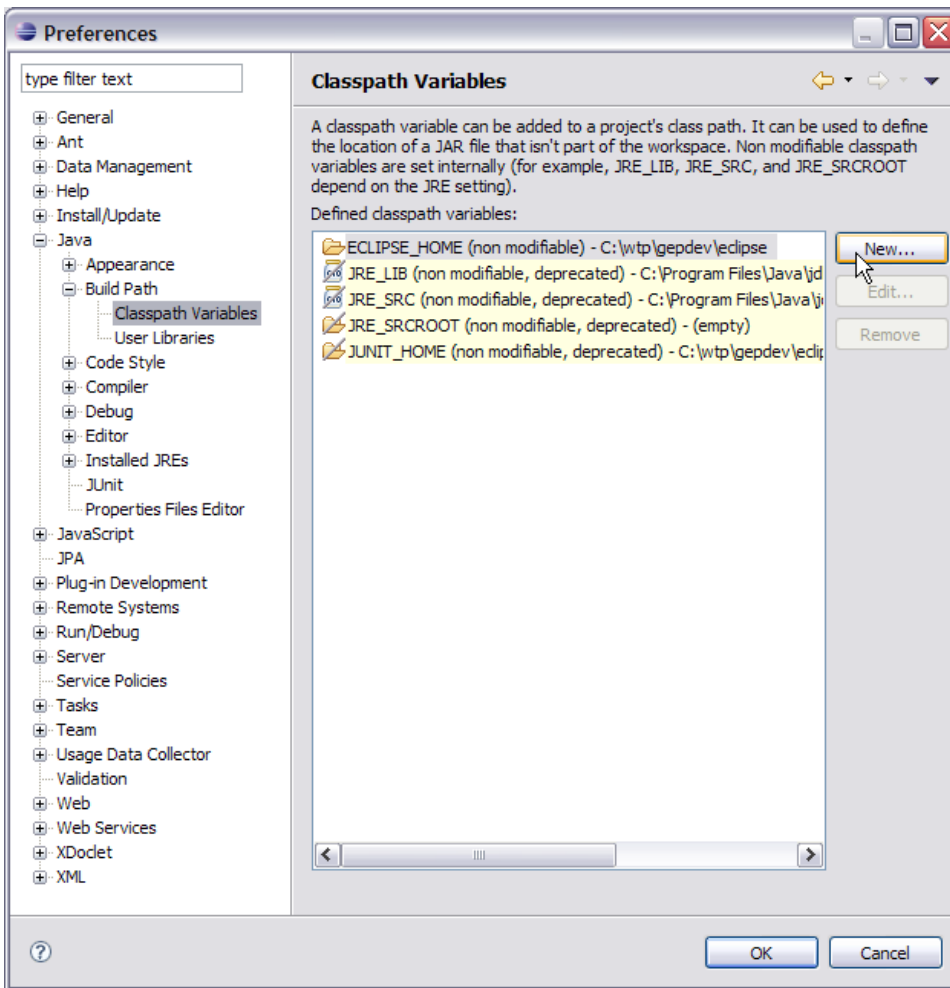
1. Build GEP using Maven. See [How to Build Geronimo Eclipse Plugin from Source](#) for details. The projects are now ready to be imported into your Eclipse workspace.
2. Launch a clean installation of Eclipse Ganymede (See [this page](#) for details).
3. Follow these steps below to complete the configuration of Eclipse for GEP development:
 - [Setting up the ClassPath variable](#)
 - [Setting up the target platform](#)
 - [Importing source code in your Eclipse Workspace](#)
 - [Run and Debug Configurations](#)

Setting up the ClassPath variable

4. Select Window->Preferences as shown in the figure.

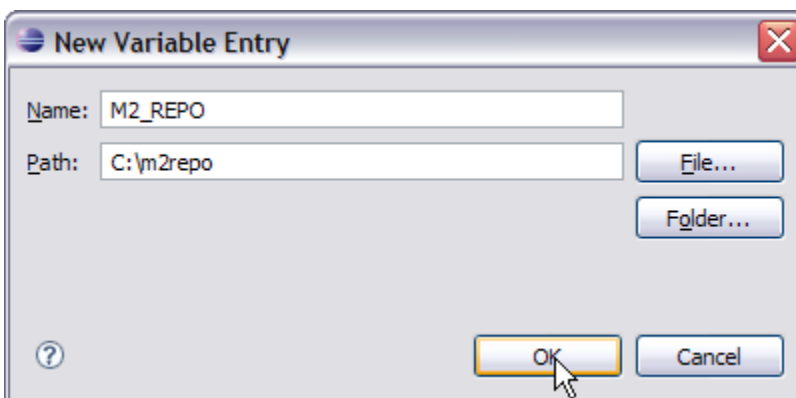


5. Select **Java->Build Path->Classpath Variables**. Select New.

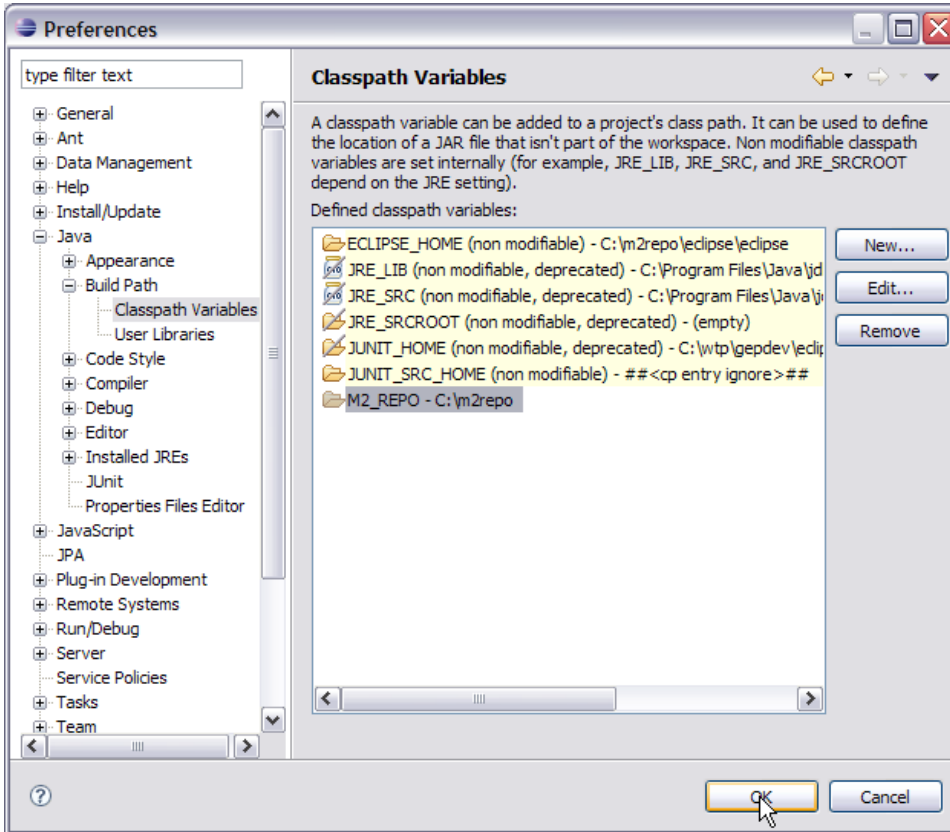


6. Enter the location of your m2repo. My m2repo is located at **C:\m2repo**. Select Ok.

```
*Name: * M2_REPO
*Path: * C:\m2repo
```



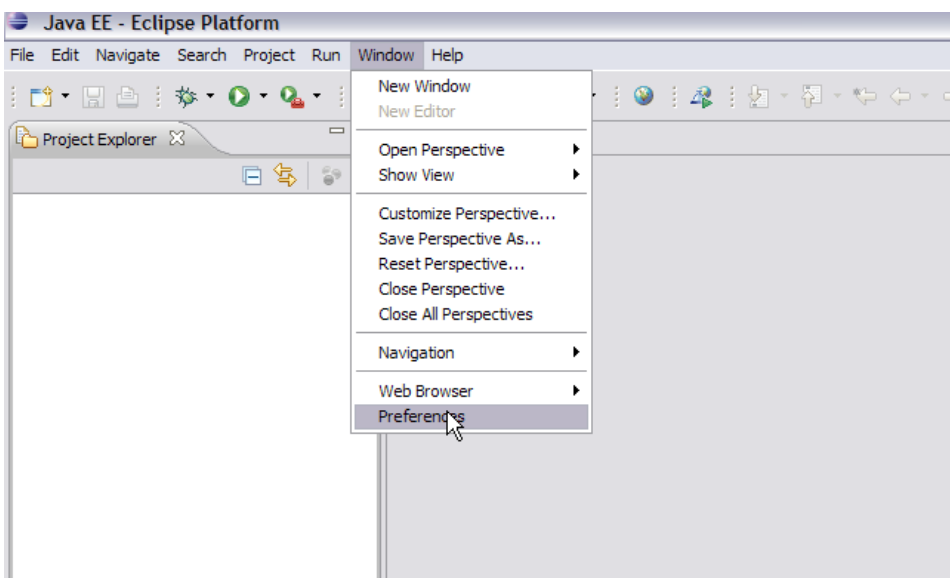
7. Once done you can see M2_REPO added to your list of ClassPath variables.



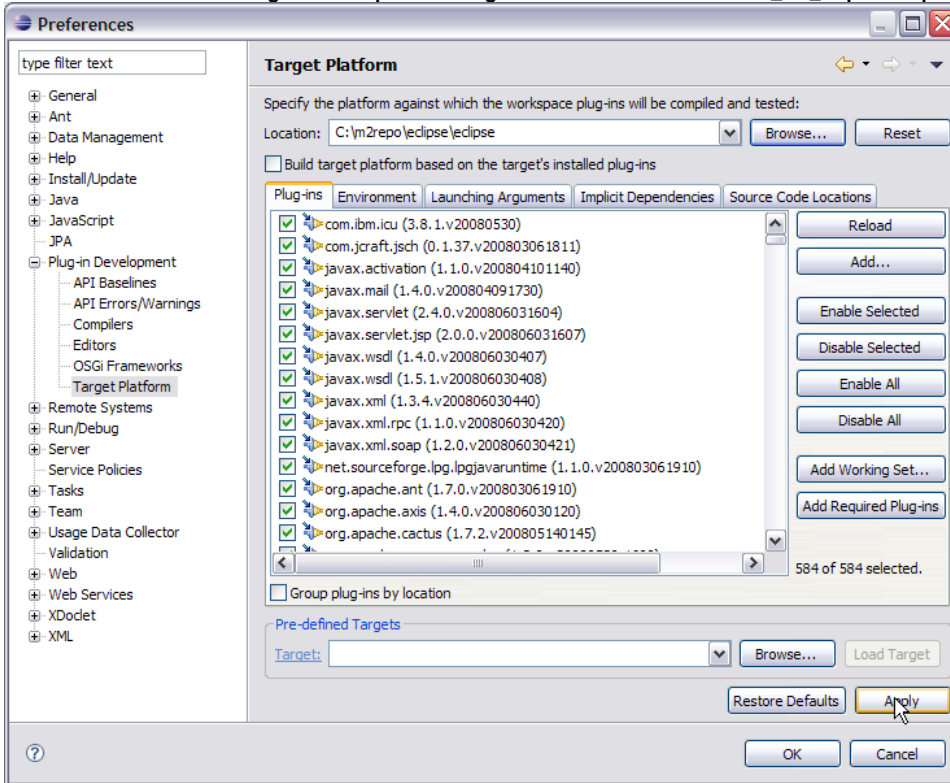
Setting up the target platform

An extract from Eclipse help about Target Platform "Target Platform refers to the platform against which you are developing and testing your plug-ins. The target platform encompasses the location of the target, the list of plug-ins constituting the target, the target environment and launching arguments." Following steps need to be performed while configuring the target platform for GEP development.

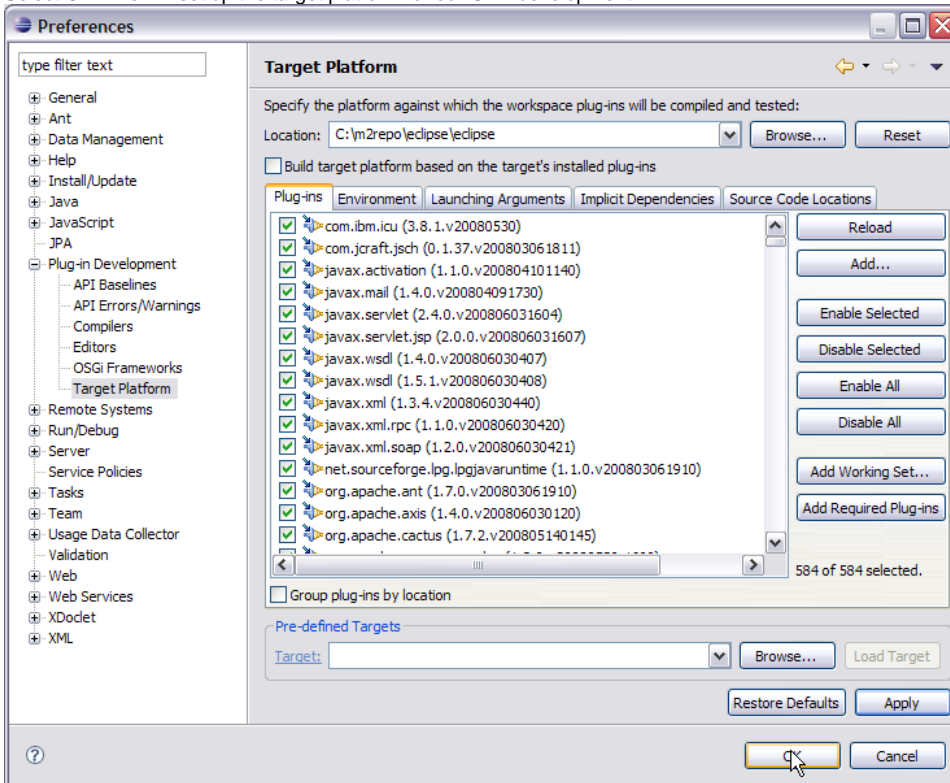
1. Select **Window->Preferences**.



2. On the next screen select **Plug-in Development->Target Platform**. Browse to **<Local_M2_Repo>/eclipse/eclipse**. Select Apply.



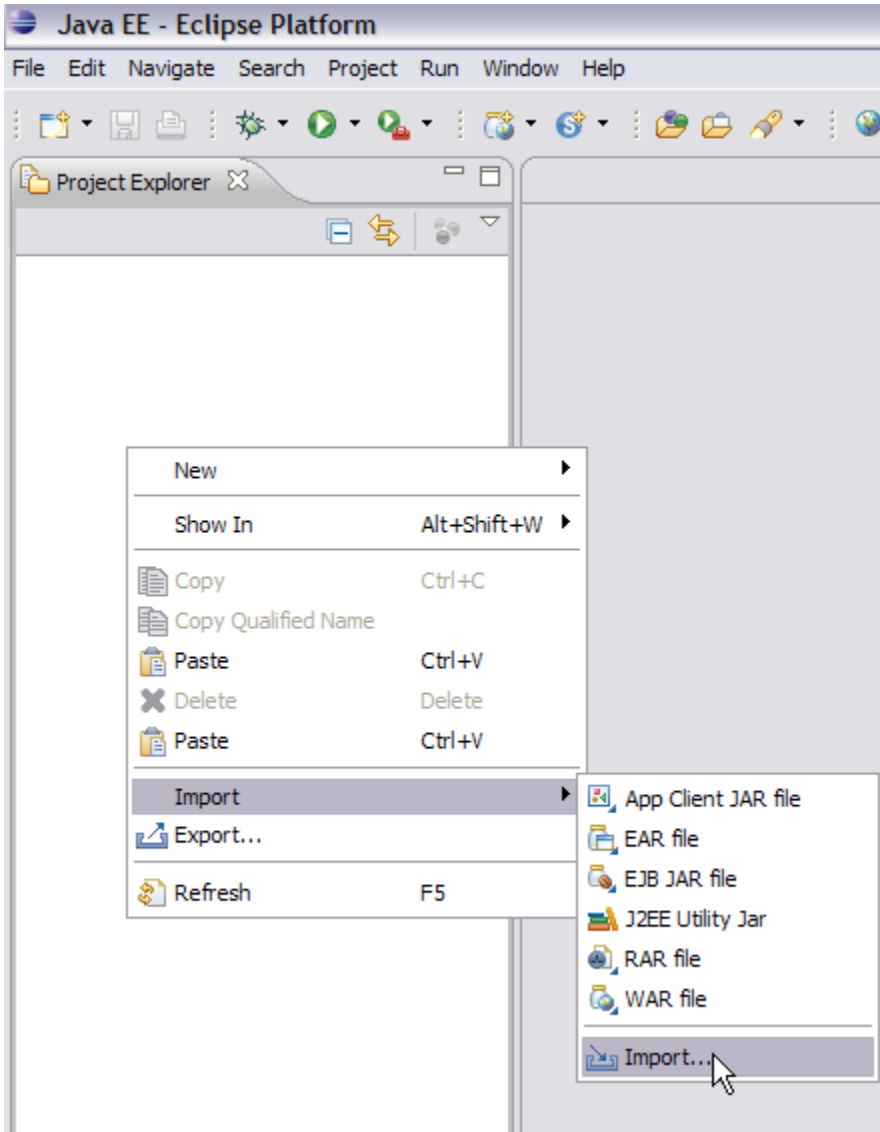
3. Select Ok. This will set up the target platform for our GEP development.



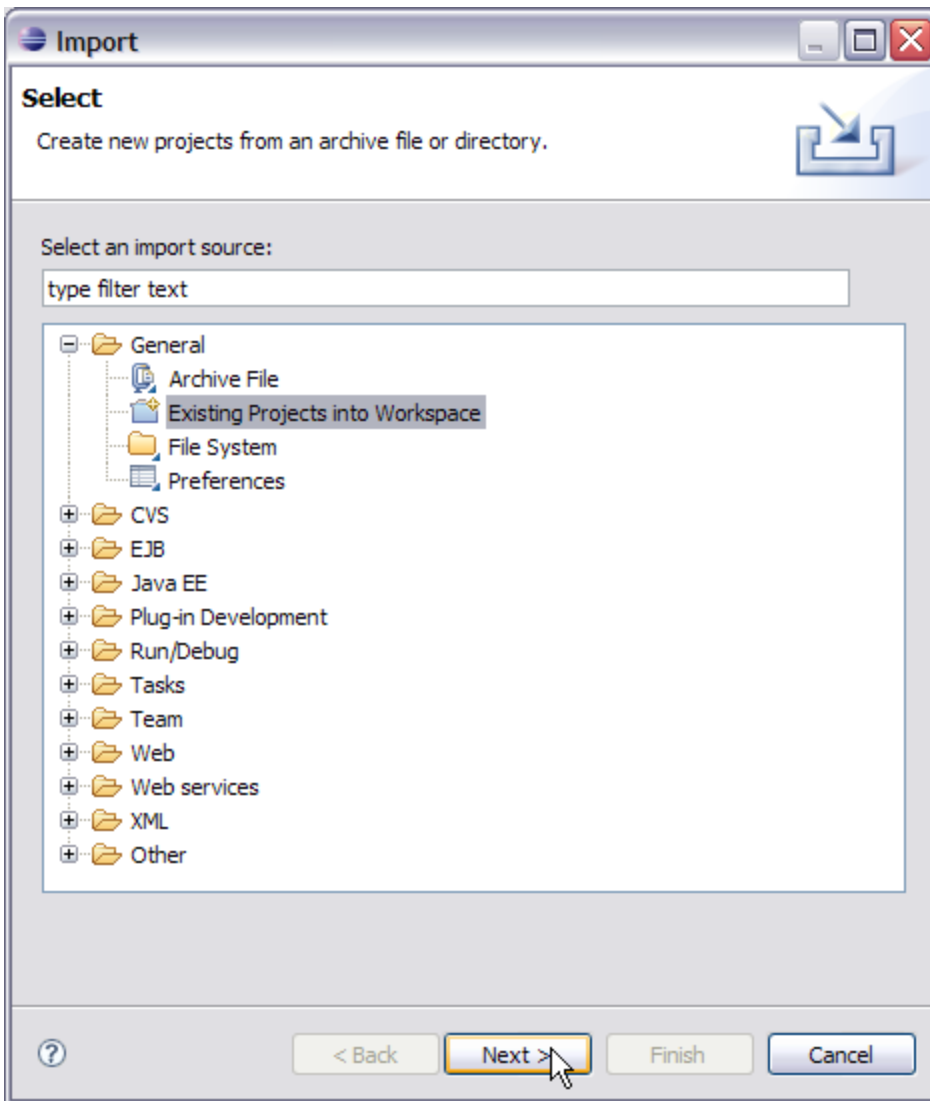
Importing source code in your Eclipse Workspace

Next step is to import the source code for plug-ins to your workspace. This is the code with which you will be playing all the time.

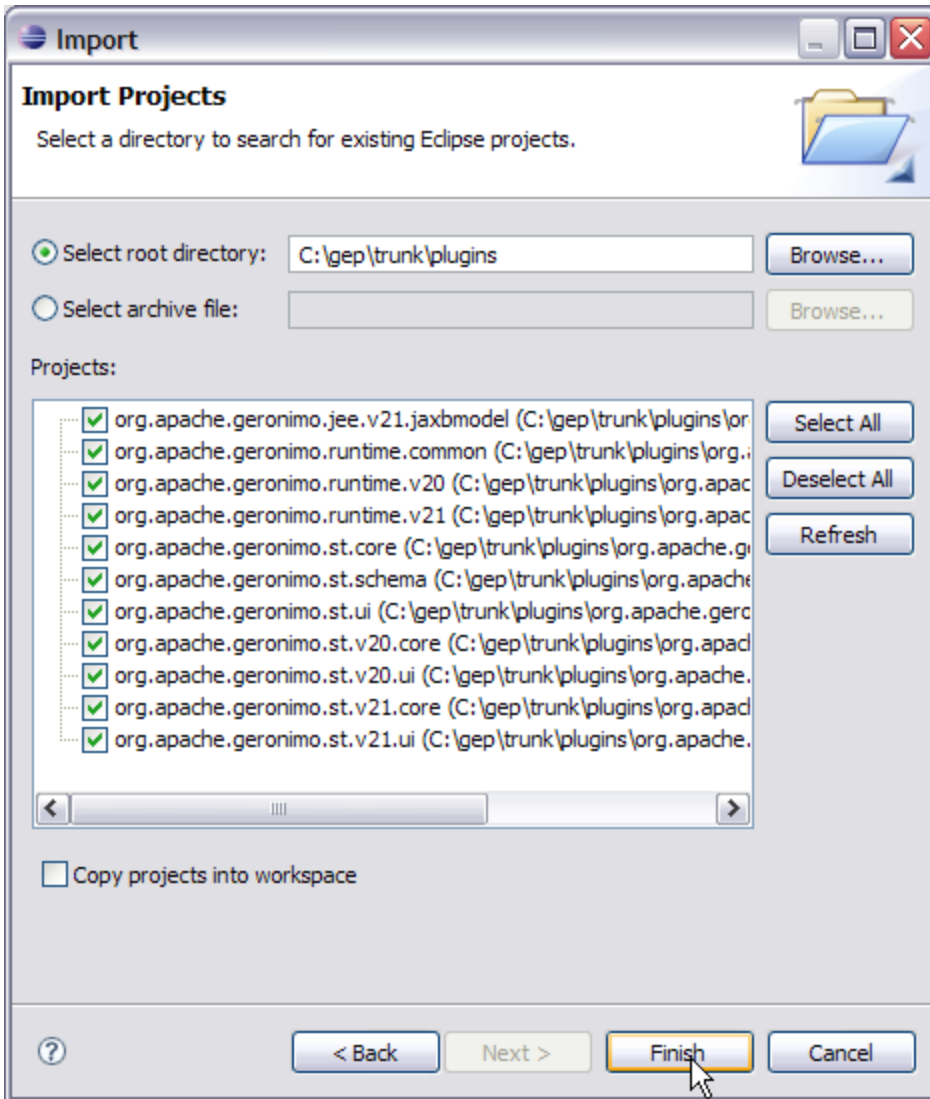
1. Right Click under **Project Explorer** and Select **Import->Import...**



2. On the next window select **General->Existing Projects into Workspace**. Select **Next**.



3. On the import projects window Browse to **<GEP_Source_Home>/trunk**. Select Finish.

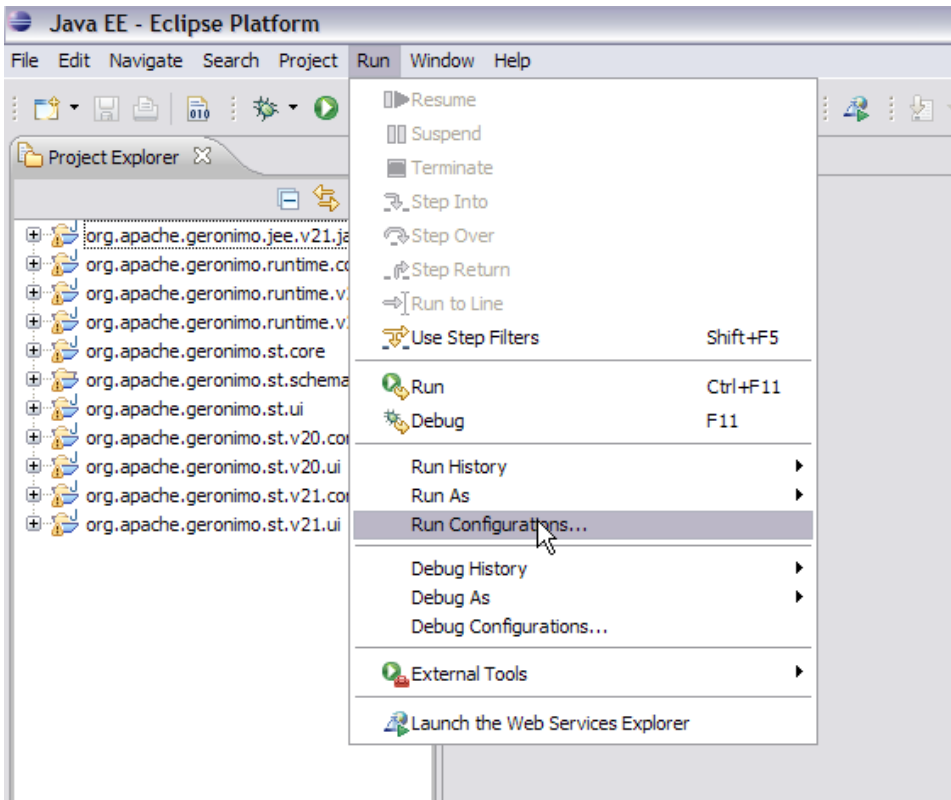


In case there are some errors try fixing those (consider it as your first step towards working with GEP! 😊) or else use the [Geronimo mailing lists](#). See [How to open a bug for Geronimo Eclipse Plugin](#) for details.

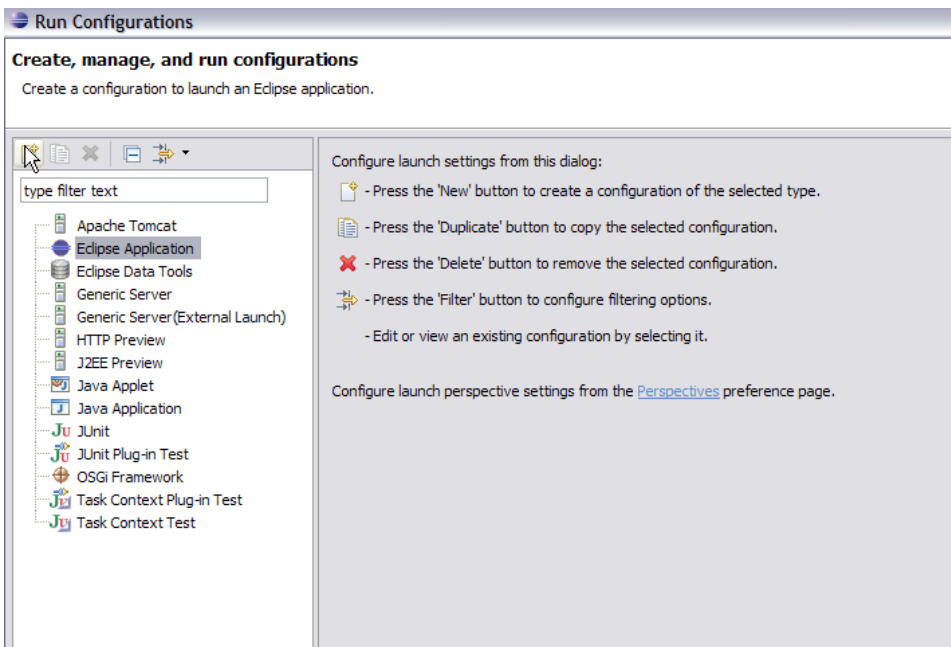
Run and Debug Configurations

Before you can start running and debusing your code, set up your debug and run configurations.

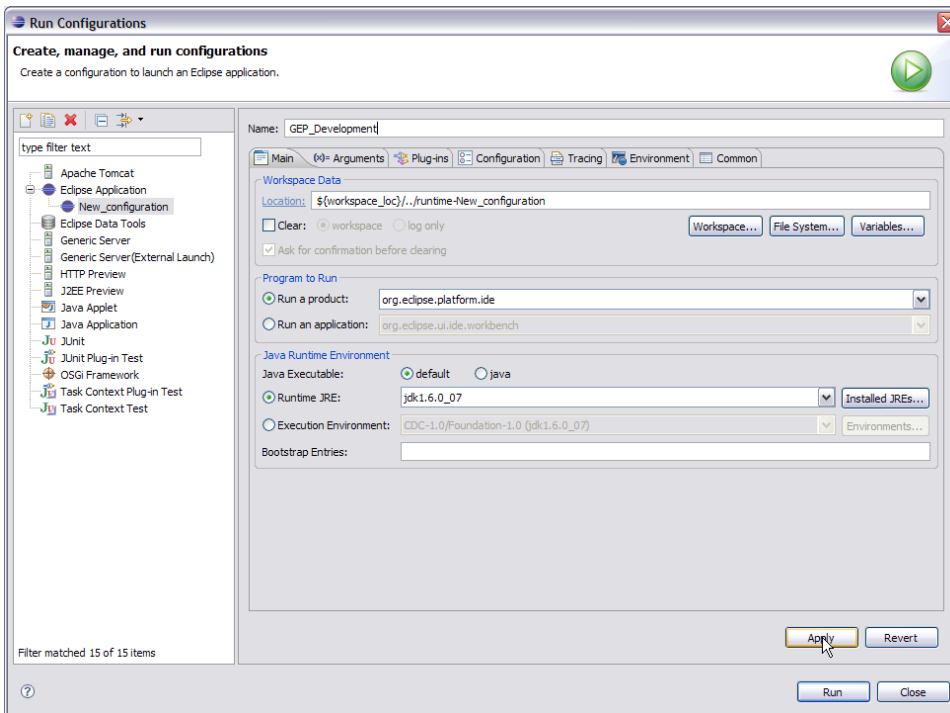
1. Select **Run->Run Configurations**.



2. On the **Run Configurations** window Select **Eclipse Application** and Select **New** button on the top left.

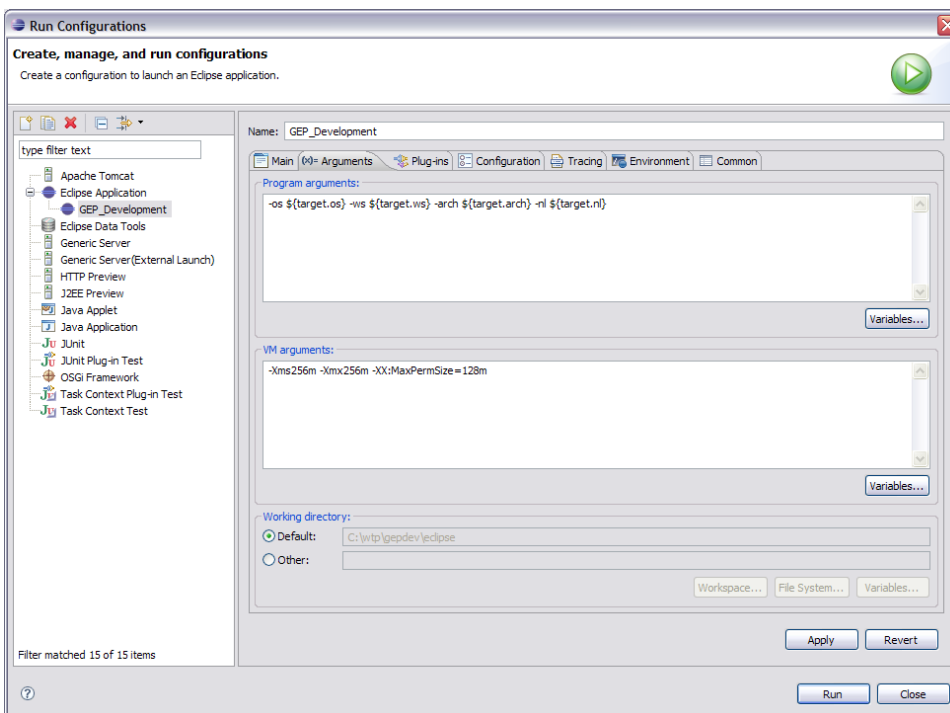


3. Name the new configuration as **GEP_DEVELOPMENT**. Also, change the workspace location name from **runtime-New_location** to **runtime-GEP_Development**. Select Apply.

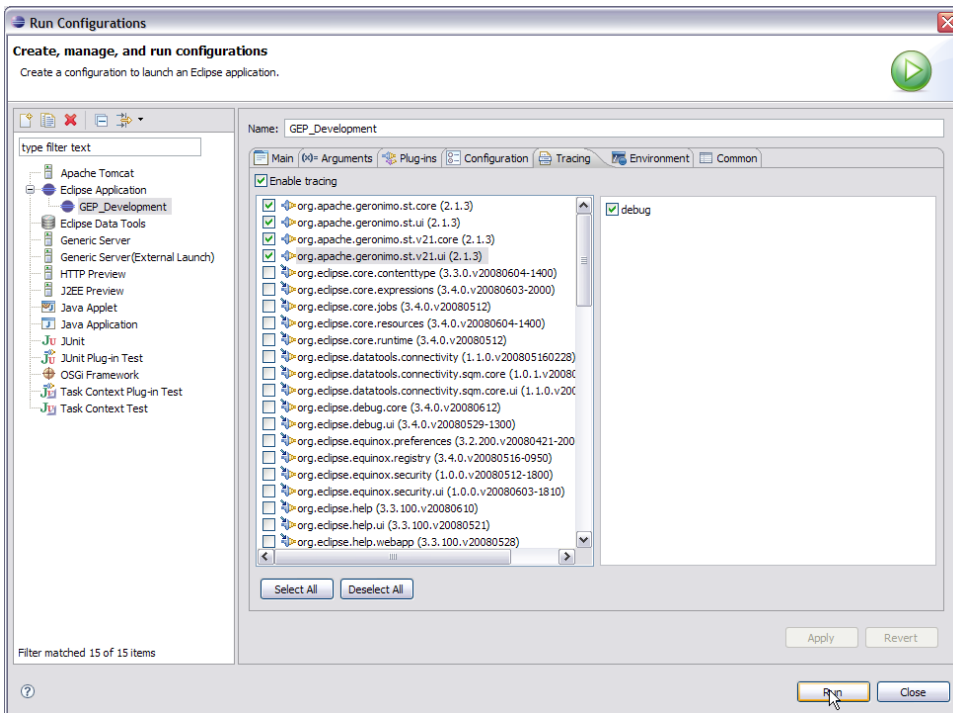


- Next select the Arguments tab for your **GEP_Development** run configuration and set the VM arguments as follows

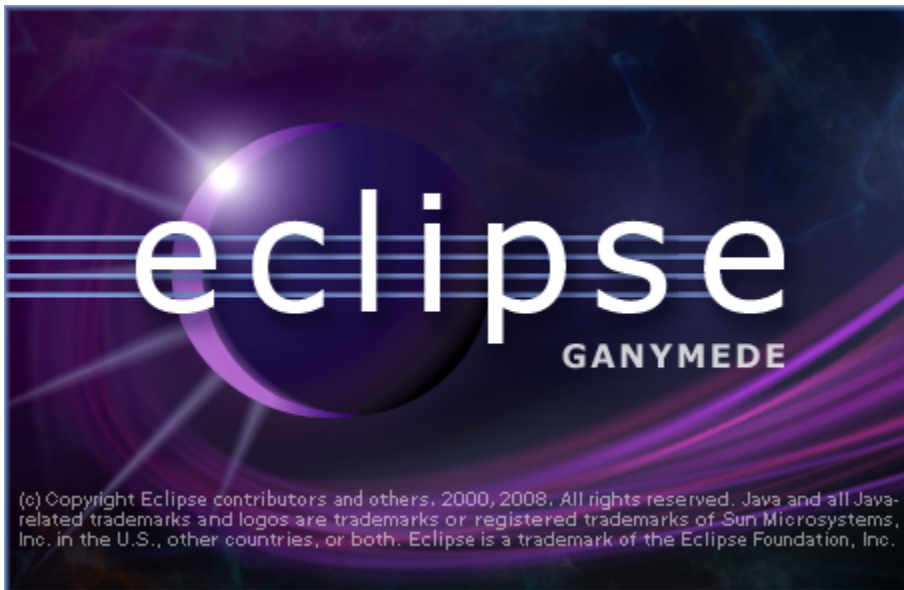
```
-Xms256m -Xmx256m -XX:MaxPermSize=128m
```



- Next select the tracing tab and select the top most 4 jars. See figure below. Select Apply and Run.



- This will launch a new Eclipse window which is nothing but **Eclipse + <Your_Plugin_Code>**. You can modify the code and each time you can launch your Eclipse Configuration to test the modifications in code.



- TO Debug the code change to Debug Perspective in eclipse and Select **Run->Debug Configurations**. Select Debug.