

OpenGTS

How to Use Eclipse with OpenGTS project

Written by Pierluigi Bucolo

V 0.0.3

Revision History

| Rev. | Date | Changed | Author |
|-------------|-------------|---|---------------|
| 0.0.1 | 26.01.09 | Initial release | PB |
| 0.0.2 | 27.11.09 | Added Windows settings and screenshots | DP |
| 0.0.3 | 05.12.90 | Added running template server inside Eclipse | OH |
| | | Debug remote tomcat application using Eclipse | PB |

Authors:

- Pierluigi Bucolo
- Daniel Poplawski
- Oskar Holowaty

How to Use Eclipse with OpenGTS project

It's possible to use Eclipse IDE with OpenGTS project, in this document I'll explain how. This document was written and tested for Linux and Windows environment (tested on Windows XP and Windows 7 x64).

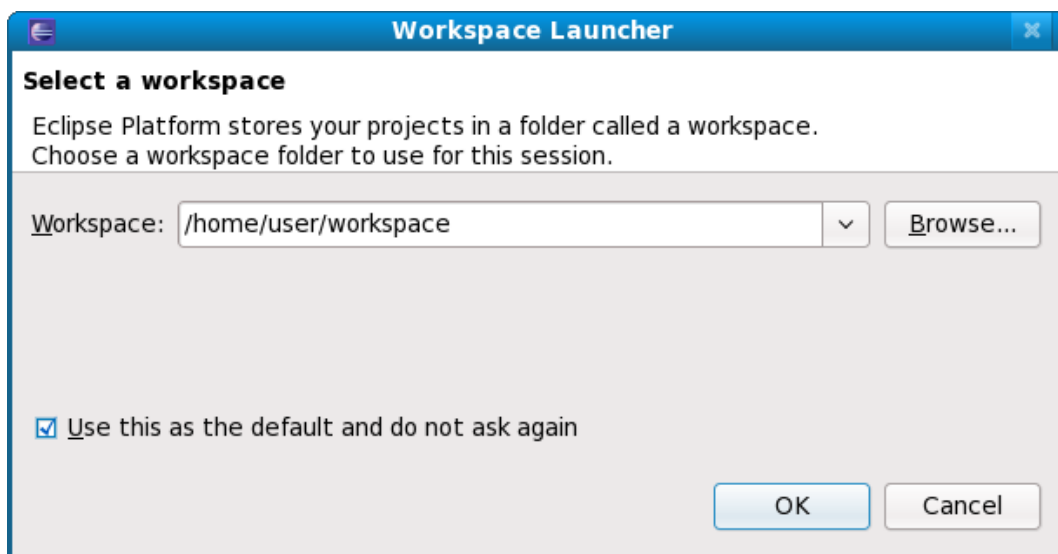
Prerequisites

The system must be setup according OpenGTS Installation/Configuration manual (OpenGTS_Config.pdf inside OpenGTS packages)
You must be able to compile correctly the package using command line before continue (ant all).

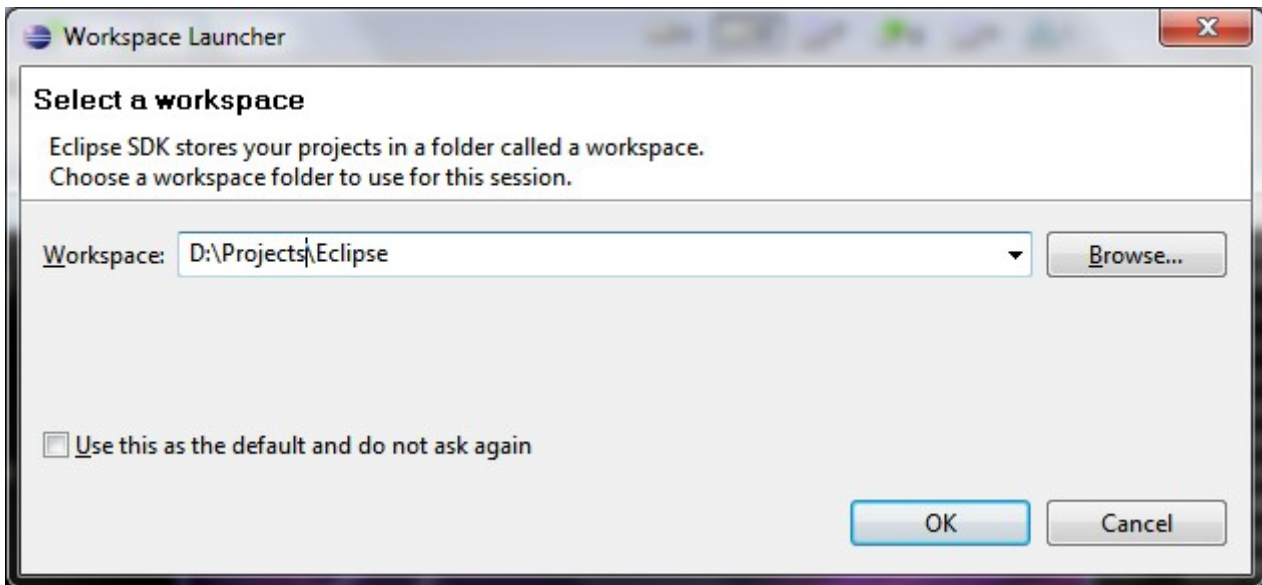
Eclipse Installation

- Download **Eclipse IDE for Java EE Developers**, this package has built in all plugins needed to develop Tomcat servlet. (<http://www.eclipse.org/downloads/>).
Linux: *eclipse-jee-ganymede-SR1-linux-gtk.tar.gz*
Windows: *eclipse-jee-ganymede-SR1-win32.zip*
- Untar or unzip the downloaded file then run **eclipse**
- When prompted select a workspace directory, then mark the checkbox to avoid to show this question every time eclipse startup.

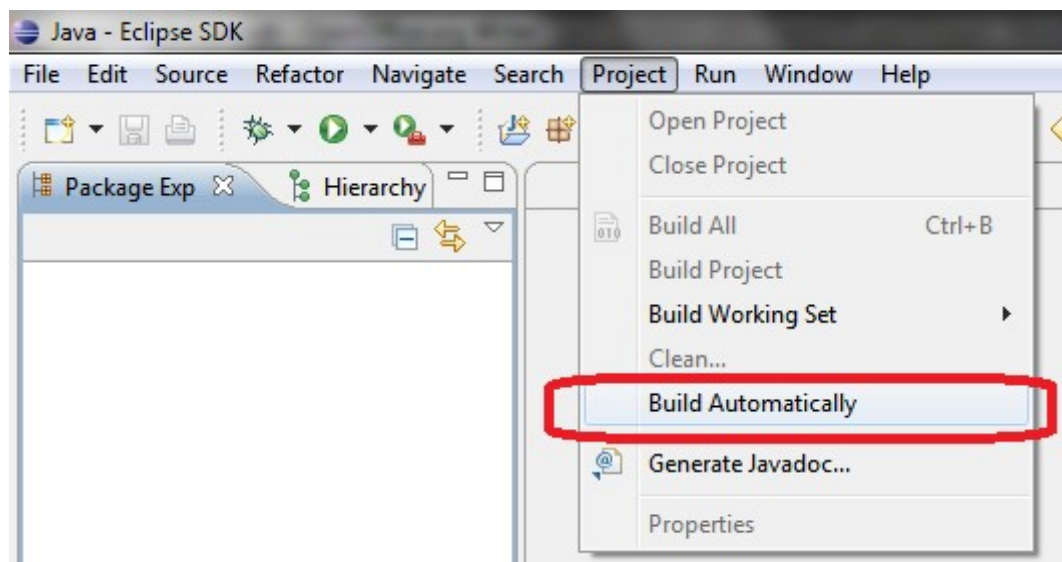
Linux: We are assuming /home/user/ is your home directory in Linux environment



Windows:

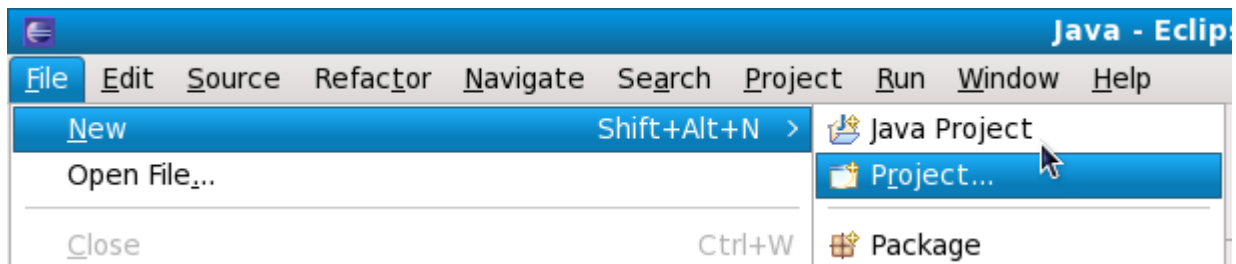


Remember to **uncheck** *Automatic Build*.

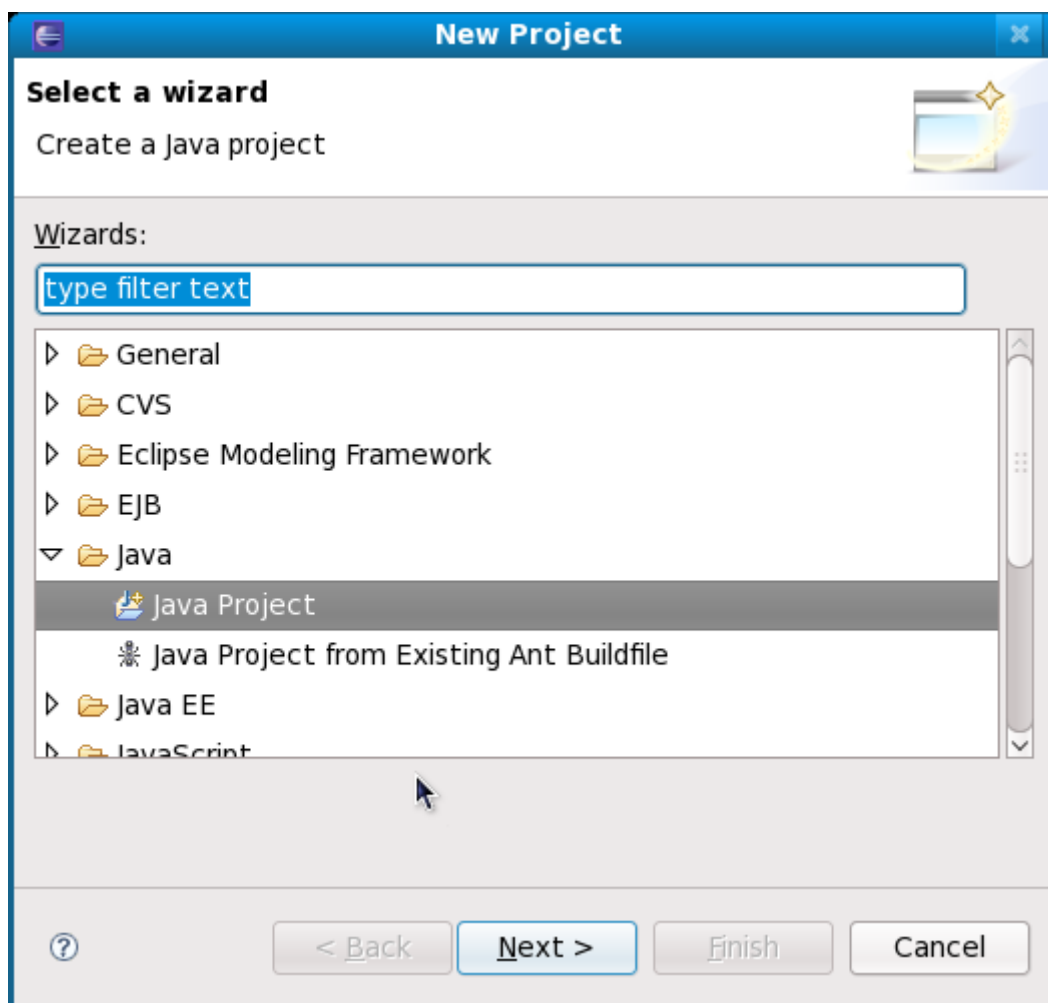


Create a new Project

- Click on **File -> New -> Project**.
According your Eclipse installation this windows maybe differ slightly



- Select **Java Project** then click **Next** button.



- Assign a name to the project: **OpenGTS**
- In *Contents* area select **Create new project in workspace**
- In *JRE* area select **Use a project specific JRE**
Be Aware!! Select ONLY SUN JDK, others doesn't works!!
Fedora starting from Release 10 uses java-1.6.0-openjdk 1.6.0-devel that works too.
- Select **Create separate folder for sources and class files**

Create a Java Project
 Create a Java project in the workspace or in an external location.

Project name:

Contents

☒ Create new project in workspace
☐ Create project from existing source

Directory:

JRE

☐ Use default JRE (Currently 'jdk1.5.0_17') [Configure JREs...](#)
☒ Use a project specific JRE:
☐ Use an execution environment JRE:

Project layout

☐ Use project folder as root for sources and class files
☒ Create separate folders for sources and class files [Configure default...](#)

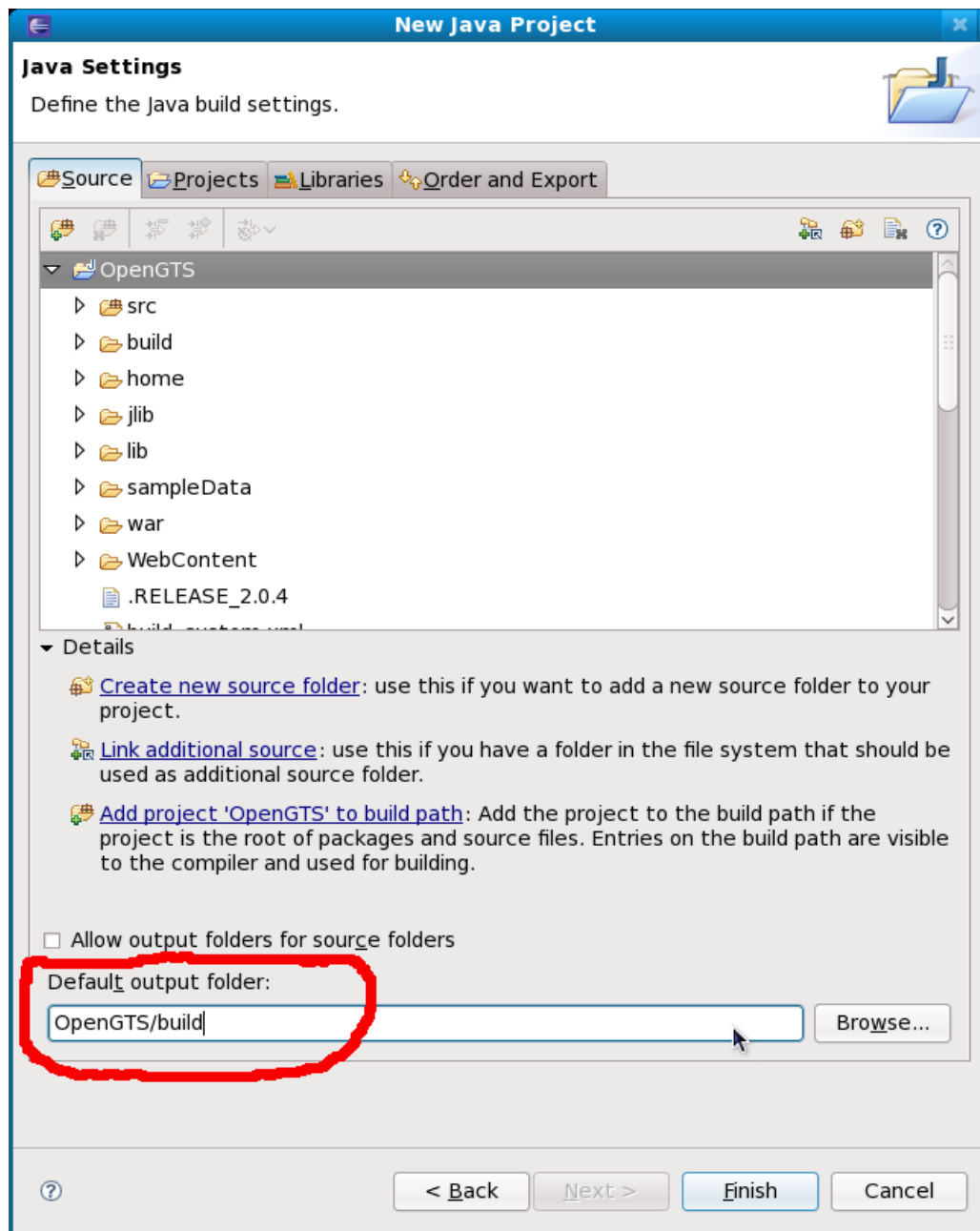
Working sets

☐ Add project to working sets

Working sets:

- Then Click **Next**.

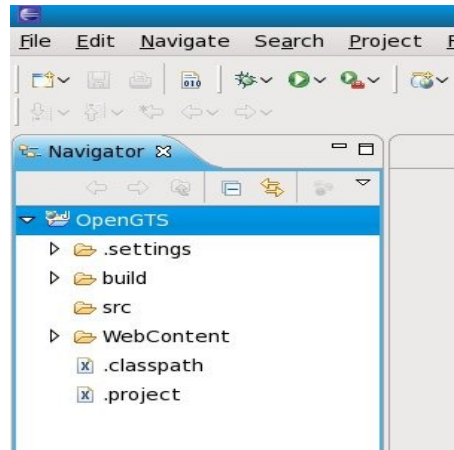
- Change *Default output folder* from *OpenGTS/bin* to **OpenGTS/build**



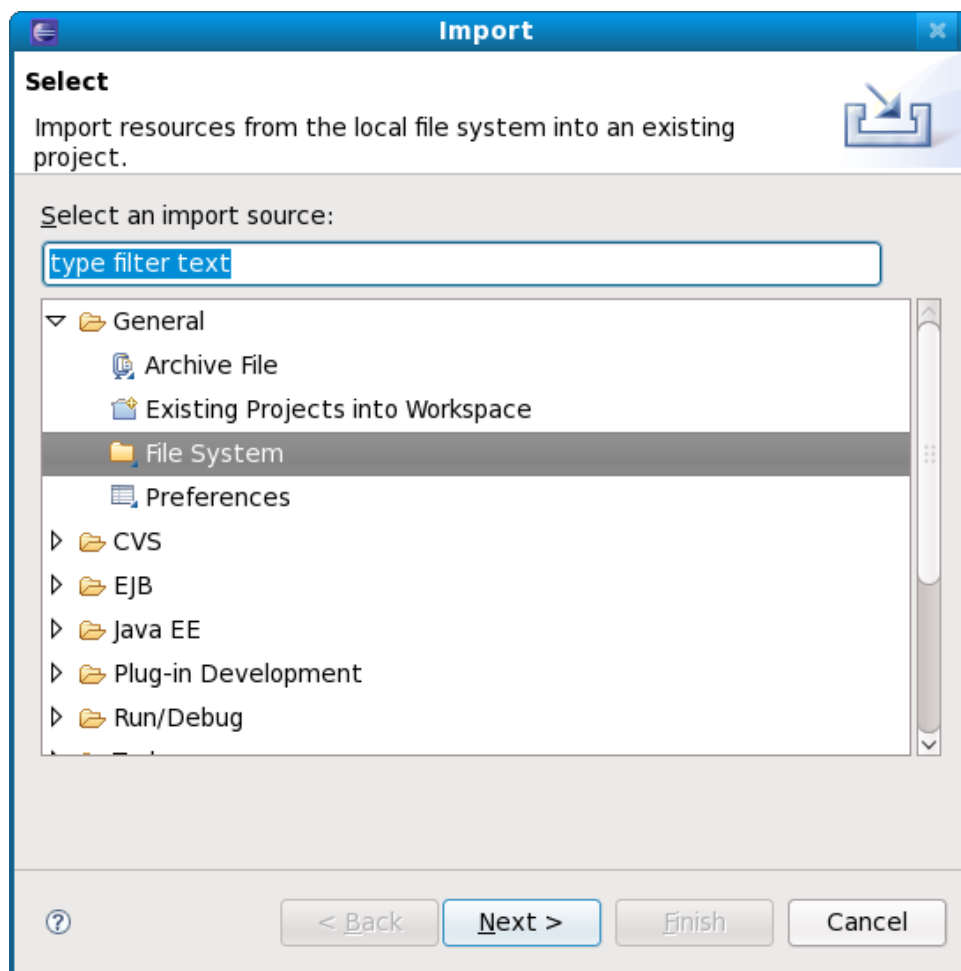
- Then click **Finish** button

Import OpenGTS Project

- Select the project just created in the Navigator pane.

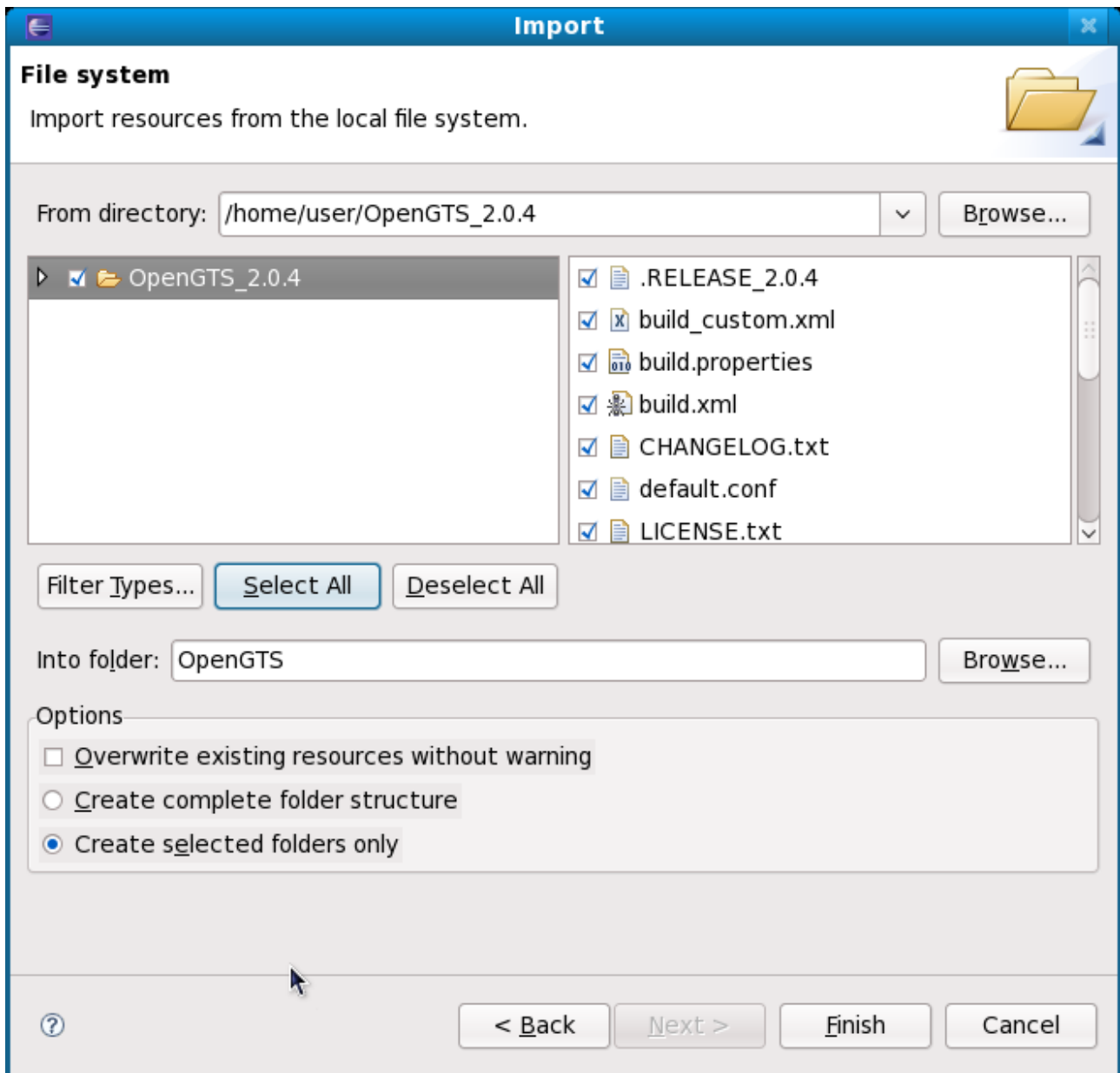


- Then click **File -> Import...**
- In the import window select **General->File System**



- Then click **Next** button

- Select the directory of OpenGTS package
- Click on the **Select All** button
- Select **Create selected folders only**

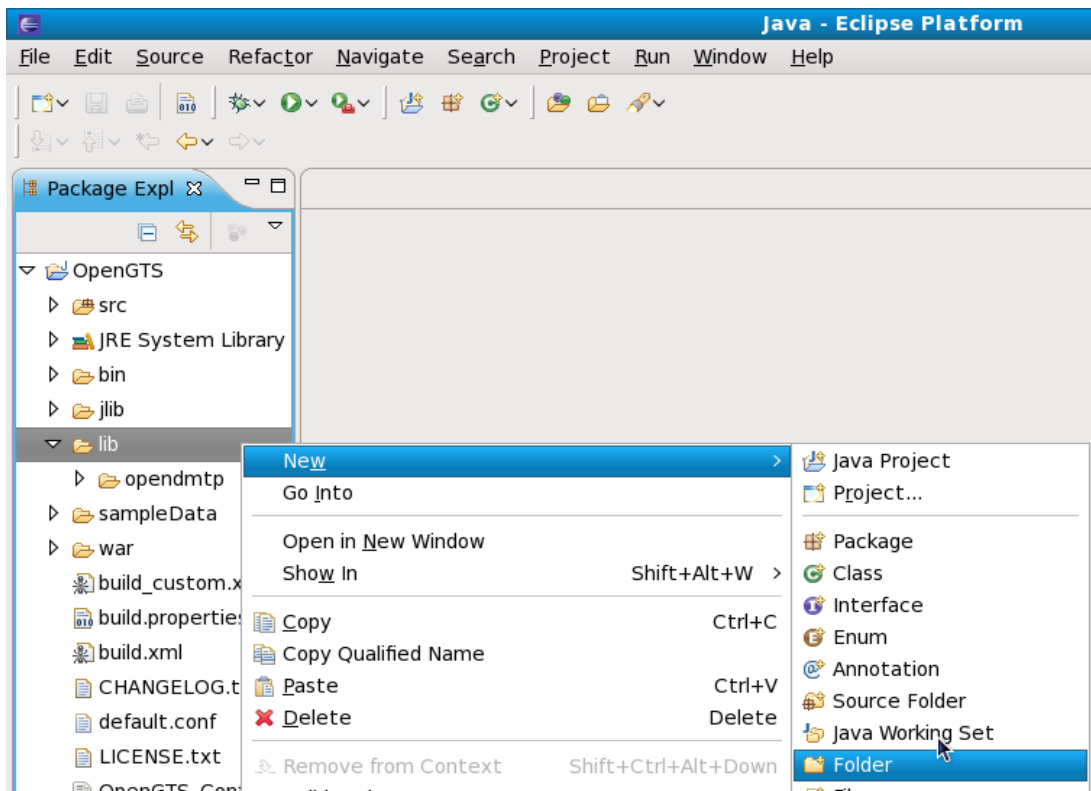


- Click on **Finish** button

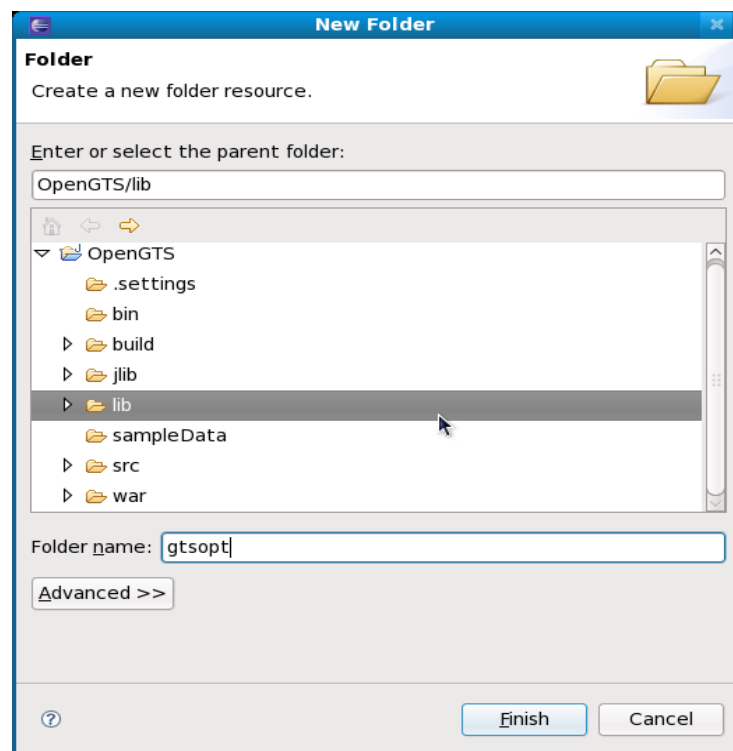
WARNING!!

Eclipse doesn't import empty folder, in OpenGTS 2.0.4 the folder **./lib/gtsopt** is empty, so we need to create it.

- Right click on the **lib** folder then select **New -> Folder**

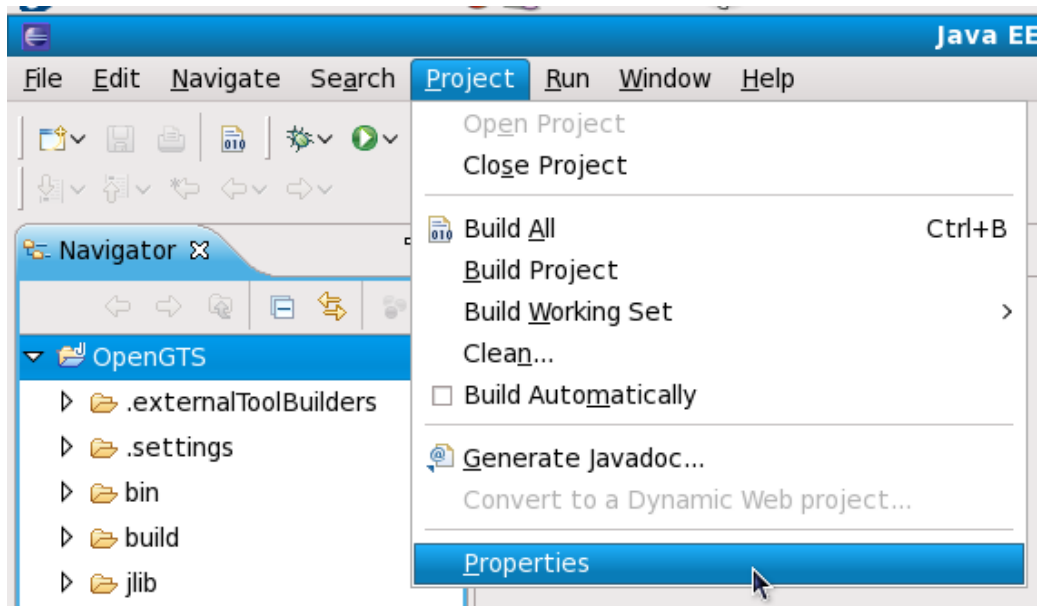


- In the *Folder name* field, type: **gtsopt**

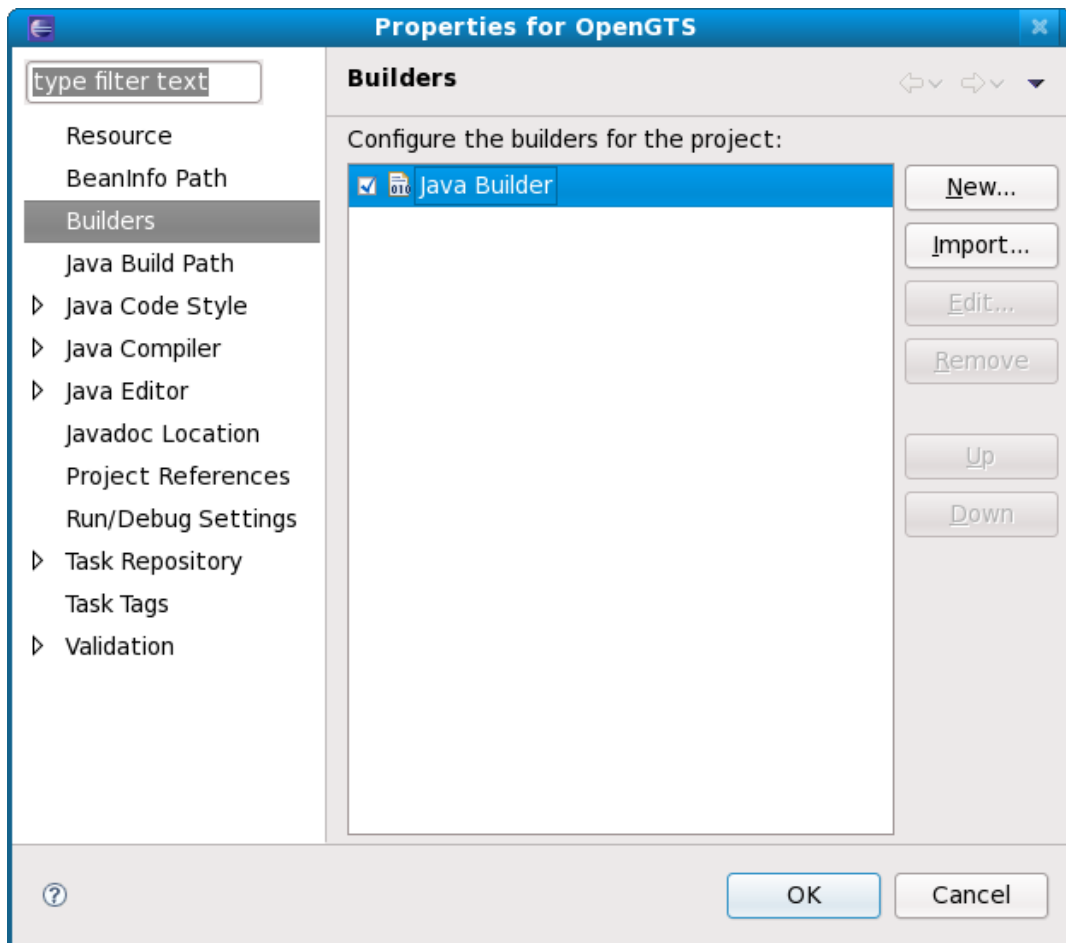


Setup Project Properties

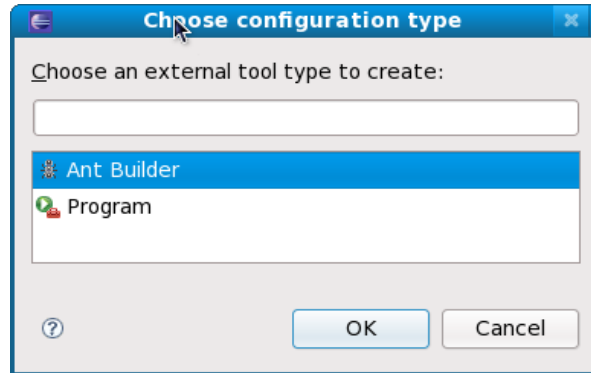
- Select in the **Navigator** pane *OpenGTS* project
- Click on menu **Project** -> **Properties**



- Select **Builders** on the left pane, then click on **New...** button

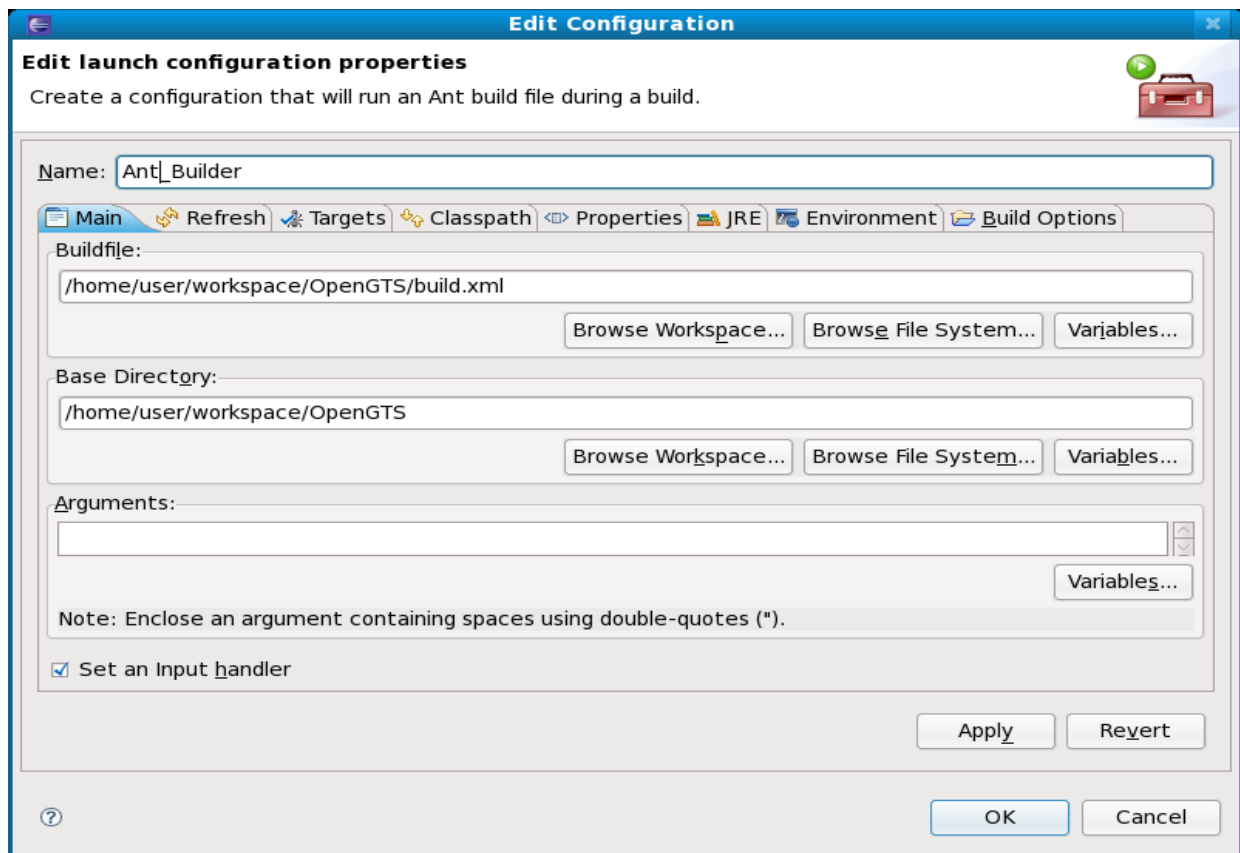


- Select **Ant Builder**, then click **OK**



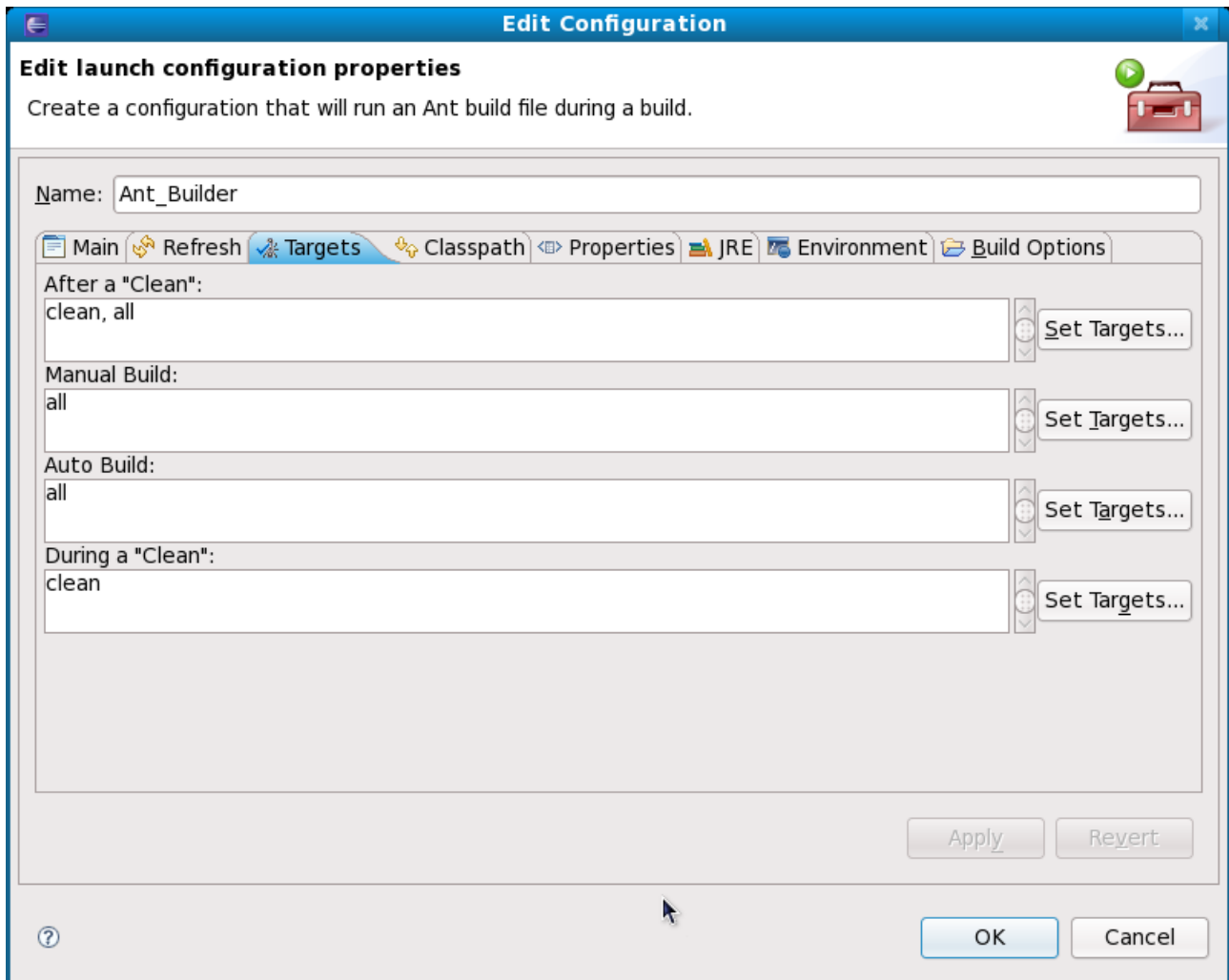
1. In the **Main** sheet complete the following fields:

- Name: Ant**
- Buildfile:** Click on **Browse File System...** button then select:
On Linux: **/home/user/workspace/OpenGTS/build.xml**
On Windows: **D:/Projects/Eclipse/OpenGTS/build.xml**
- Base Directory:** Click on **Browse File System...** button then select
On Linux: **home/user/workspace/OpenGTS/**
On Windows: **D:/Projects/Eclipse/OpenGTS/**
- Click on **Apply**



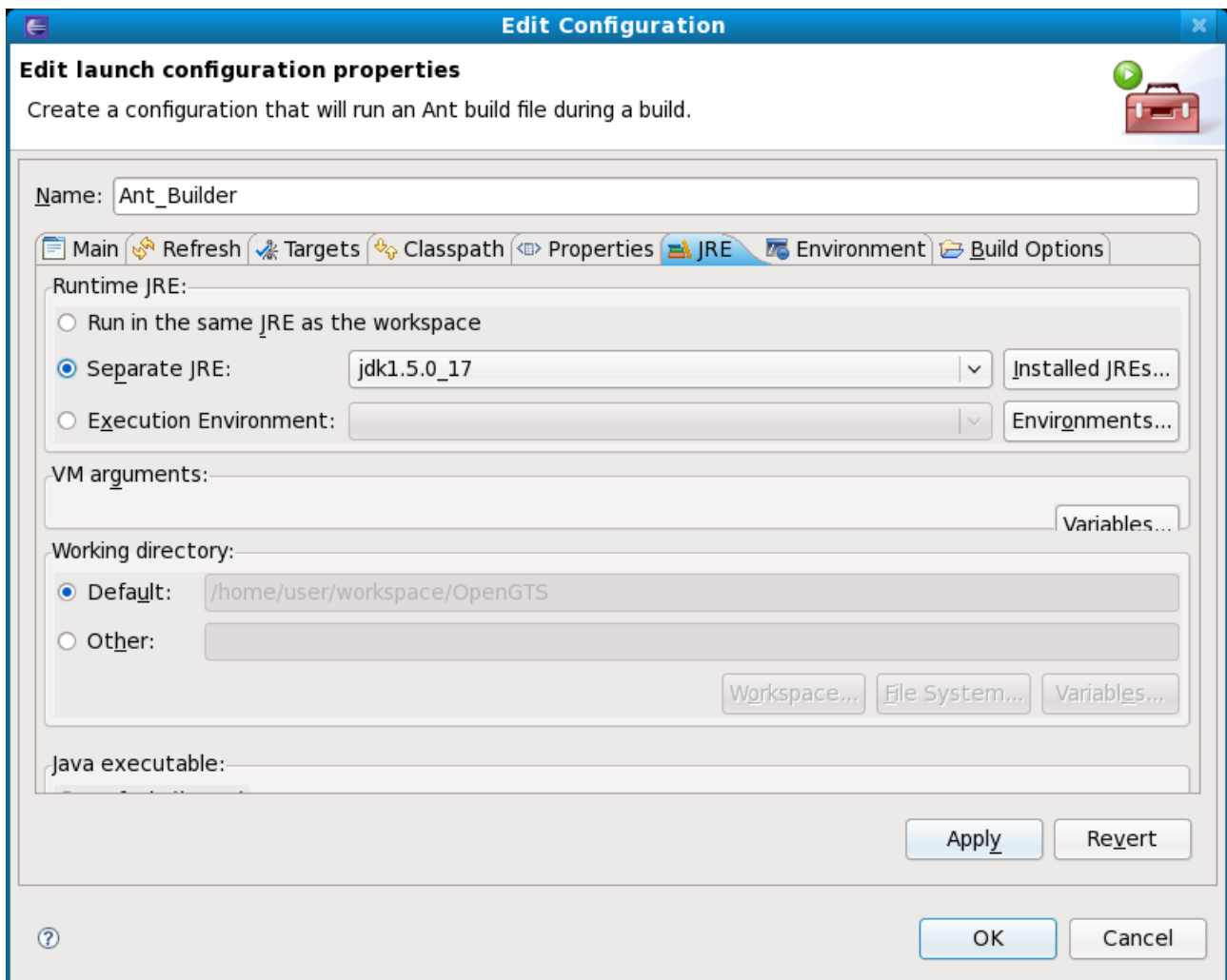
2. In the **Targets** sheet complete the following fields:

- a) *After a "Clean"*: Click on **Set Targets...** and select **clean** and **all**
(WARNING!! Check the right order!!).
- b) *Manual build*: Click on **Set Targets...** and select **all**.
- c) *Auto Build*: Click on **Set Targets...** and select **all**.
- d) *During a "Clean"*: Click on **Set Targets...** and select **clean**.
- e) Click on **Apply** button.



3. In the **JRE** sheet complete the fields according the picture below:

ATTENTION: If you use different JDK version, select the SUN JDK, other doesn't works!
(on Fedora you can use openjdk package, see note before).



4. In the **Environment** sheet:

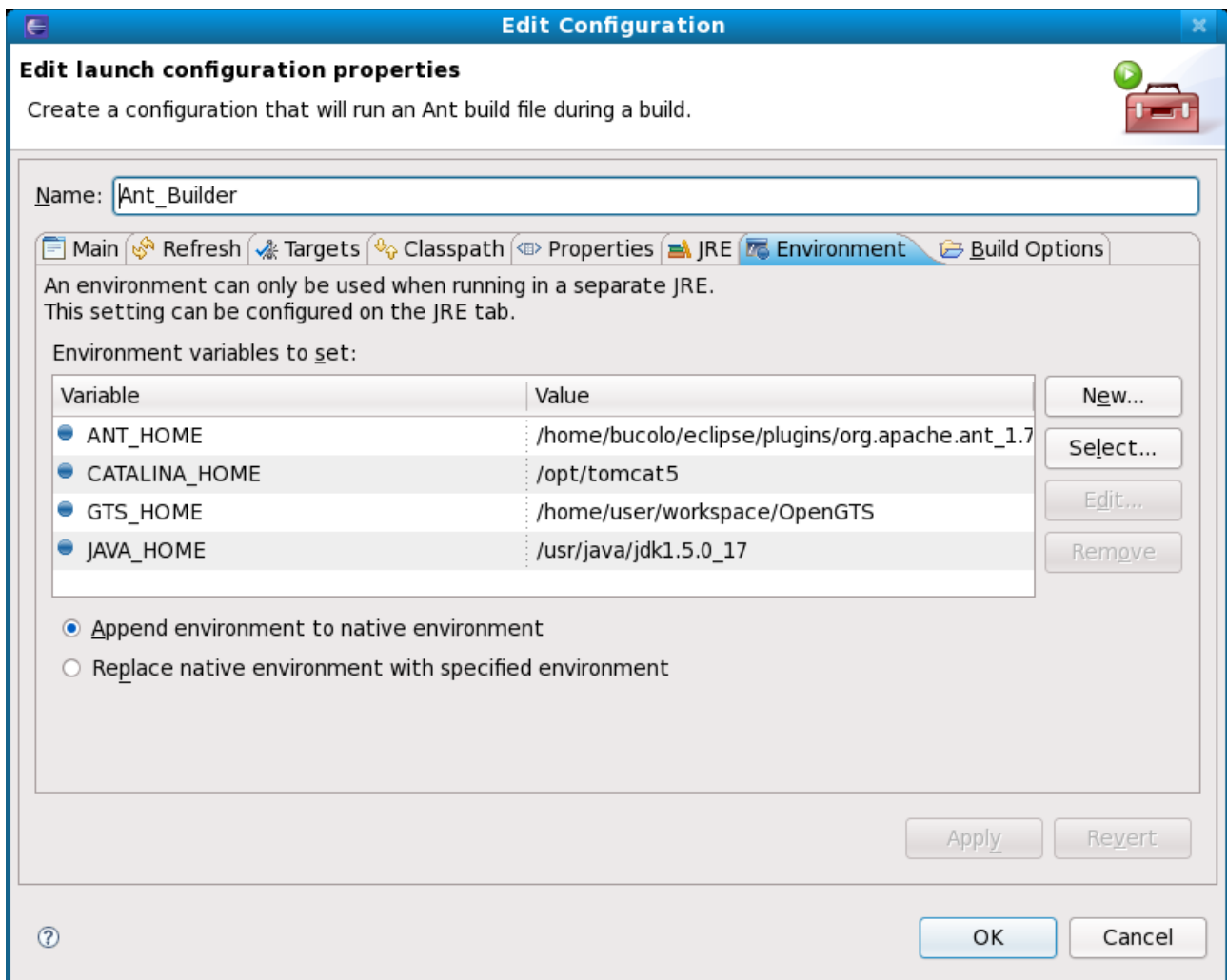
a) Click on **New...** button and insert the variables:

- **CATALINA_HOME**
- **JAVA_HOME**
- **GTS_HOME**
- **ANT_HOME (optional)**

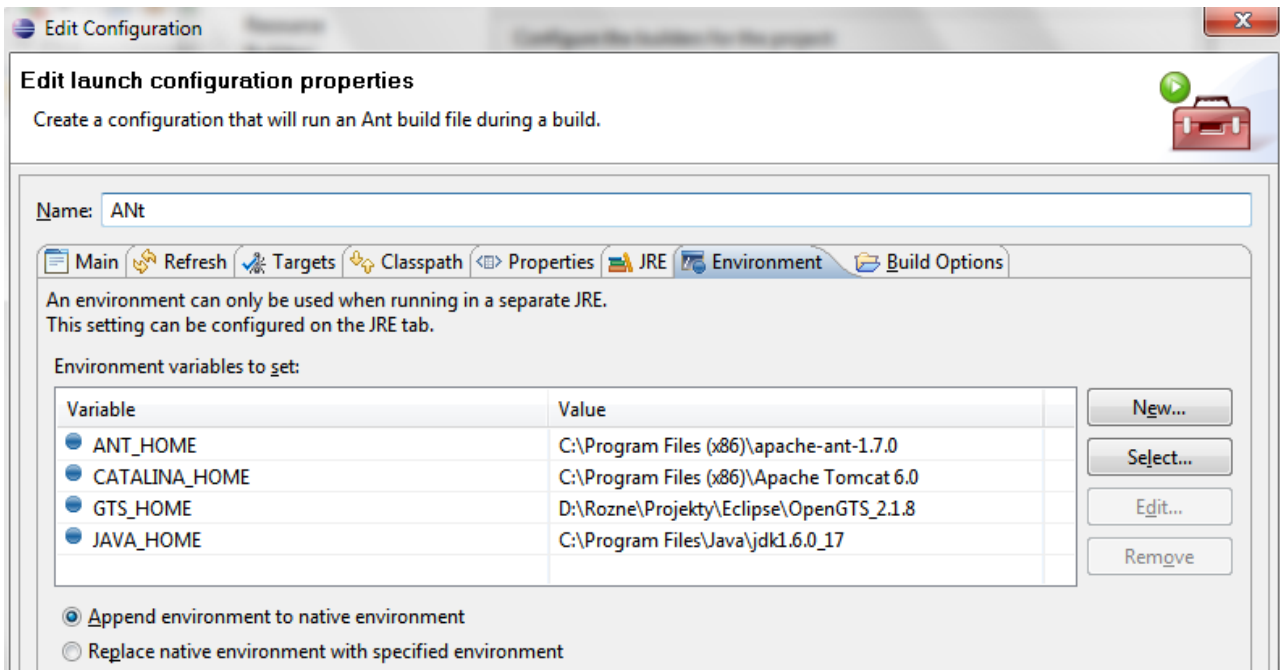
TIPS: You don't need to create ANT_HOME, because is added by Eclipse after first build

b) Select **Append environment to native environment**

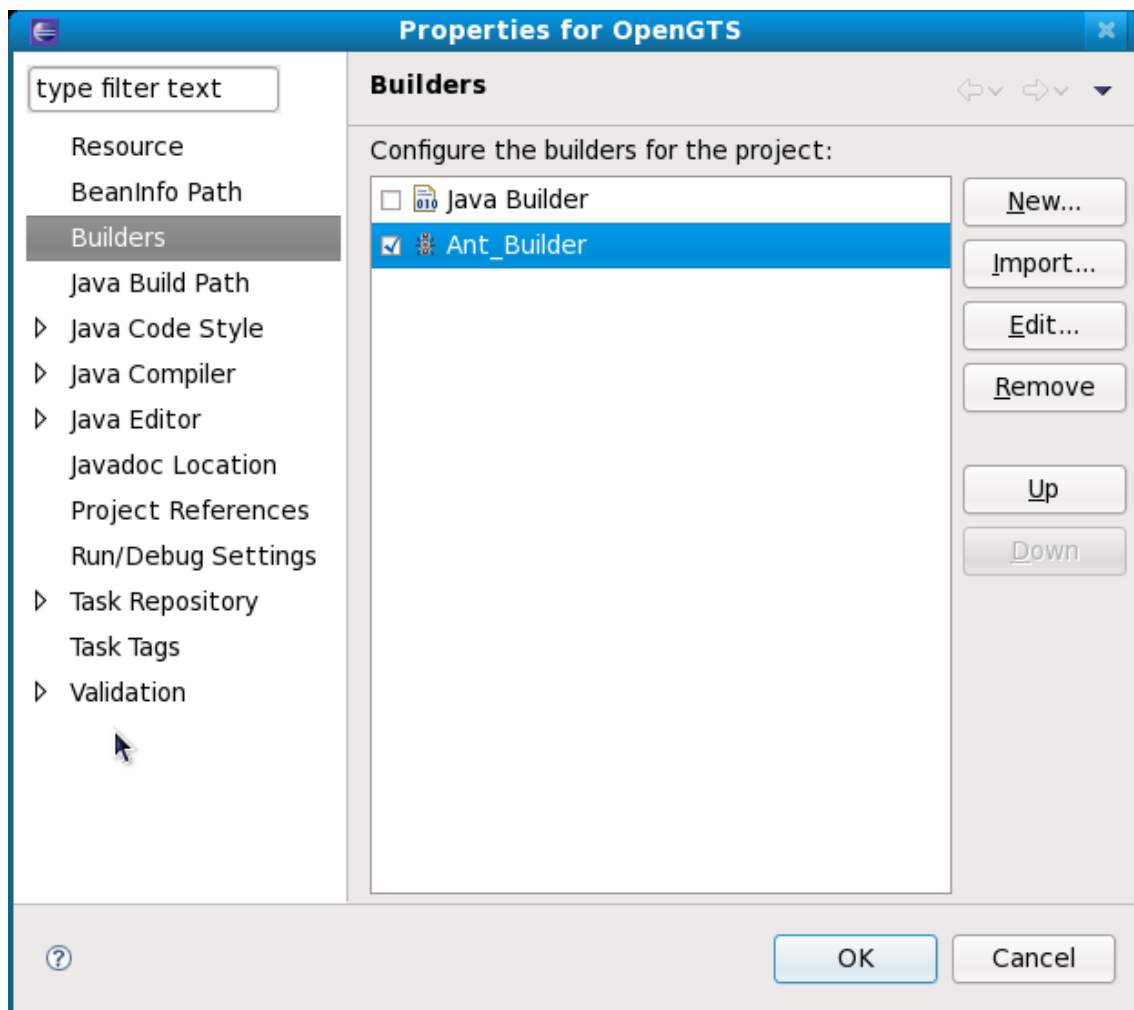
c) Click on **OK**



On Windows:

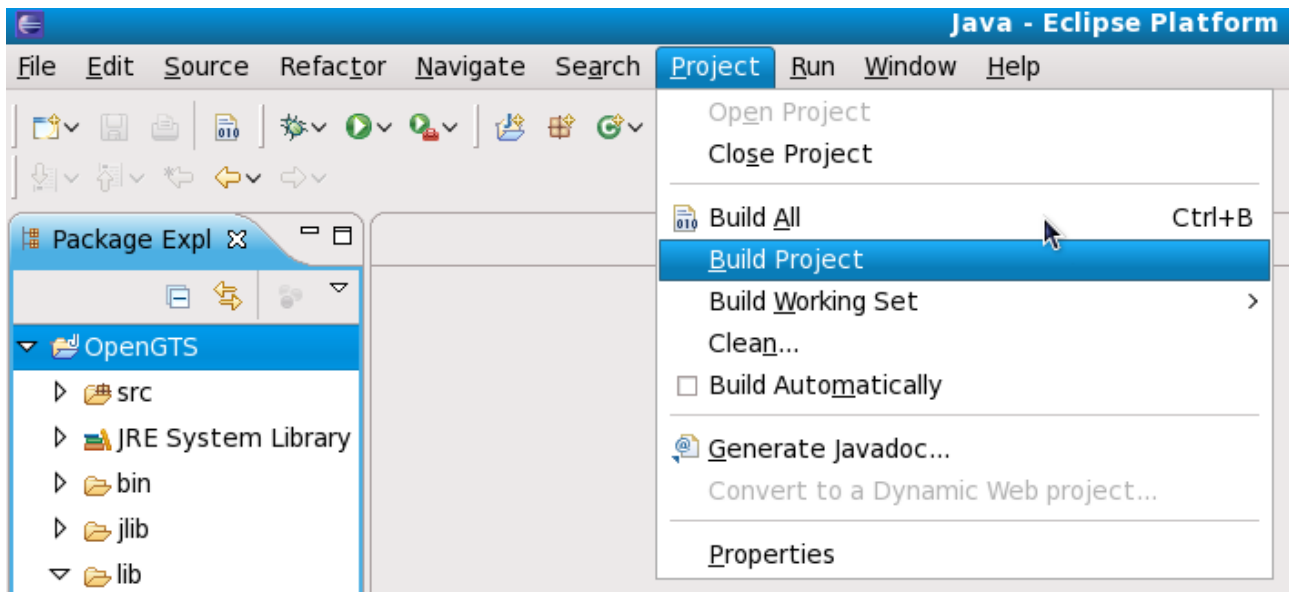


- Unselect **Java Builder** then click **OK**

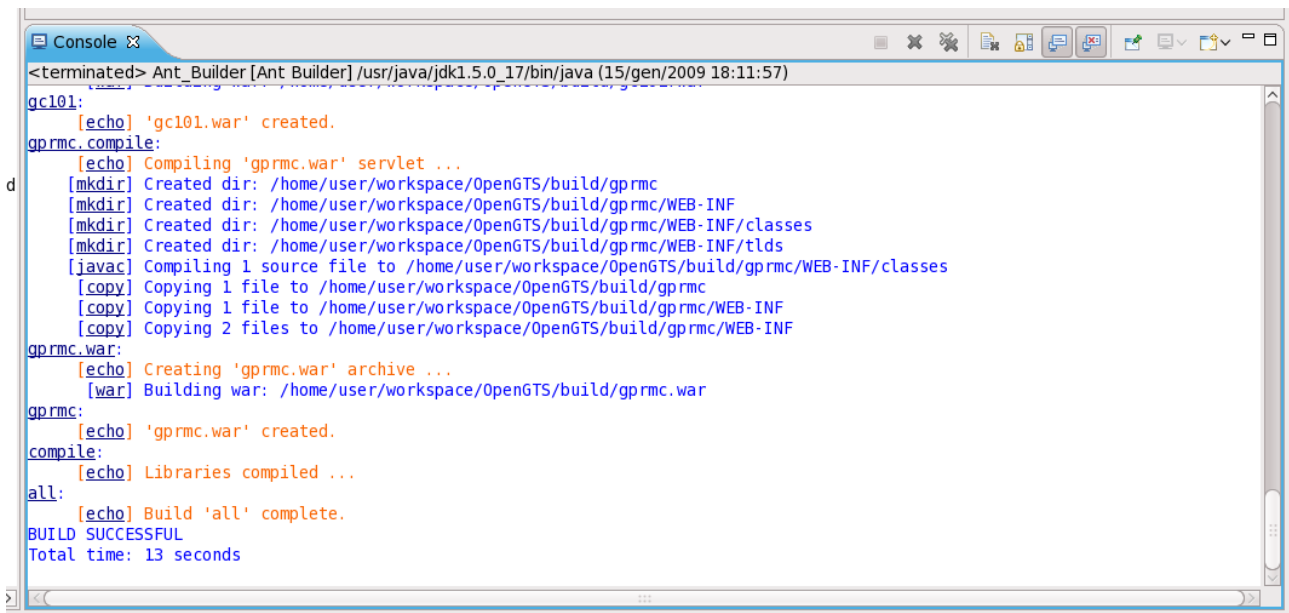


Compile OpenGTS

- Click on **Project -> Build Project**



- In the **Console** pane, you must read after some printed rows: **BUILD SUCCESSFUL**



On Windows:

```
<terminated> ANt [Ant Builder] C:\Program Files\Java\jre6\bin\javaw.exe (2009-11-27 21:20:33)
gprmc:
    [echo] 'gprmc.war' created.
gpsmapper.compile:
    [echo] Compiling 'gpsmapper.war' servlet ...
    [mkdir] Created dir: D:\Rozne\Projekty\Eclipse\OpenGTS_2.1.8\build\gpsmapper
    [mkdir] Created dir: D:\Rozne\Projekty\Eclipse\OpenGTS_2.1.8\build\gpsmapper\WEB
    [mkdir] Created dir: D:\Rozne\Projekty\Eclipse\OpenGTS_2.1.8\build\gpsmapper\WEB
    [mkdir] Created dir: D:\Rozne\Projekty\Eclipse\OpenGTS_2.1.8\build\gpsmapper\WEB
    [javac] Compiling 1 source file to D:\Rozne\Projekty\Eclipse\OpenGTS_2.1.8\build
    [copy] Copying 1 file to D:\Rozne\Projekty\Eclipse\OpenGTS_2.1.8\build\gpsmappe
    [copy] Copying 5 files to D:\Rozne\Projekty\Eclipse\OpenGTS_2.1.8\build\gpsmapp
    [copy] Copying 2 files to D:\Rozne\Projekty\Eclipse\OpenGTS_2.1.8\build\gpsmapp
gpsmapper.war:
    [echo] Creating 'gpsmapper.war' archive ...
    [war] Building war: D:\Rozne\Projekty\Eclipse\OpenGTS_2.1.8\build\gpsmapper.wa
gpsmapper:
    [echo] 'gpsmapper.war' created.
compile.servlets:
    [echo] Servlet/War file compiled ...
compile:
    [echo] Libraries compiled ...
all:
    [echo] Build 'all' complete.
BUILD SUCCESSFUL
Total time: 2 minutes 52 seconds
```

NOTE:

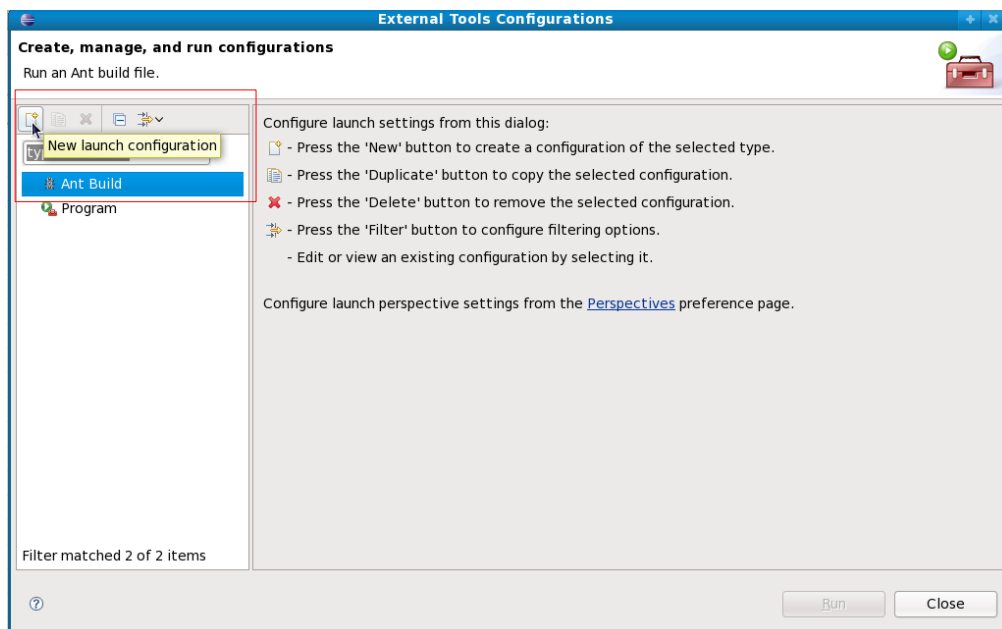
Remember that Eclipse don't want to build again if you didn't make any changes to source code.

Run template server inside Eclipse

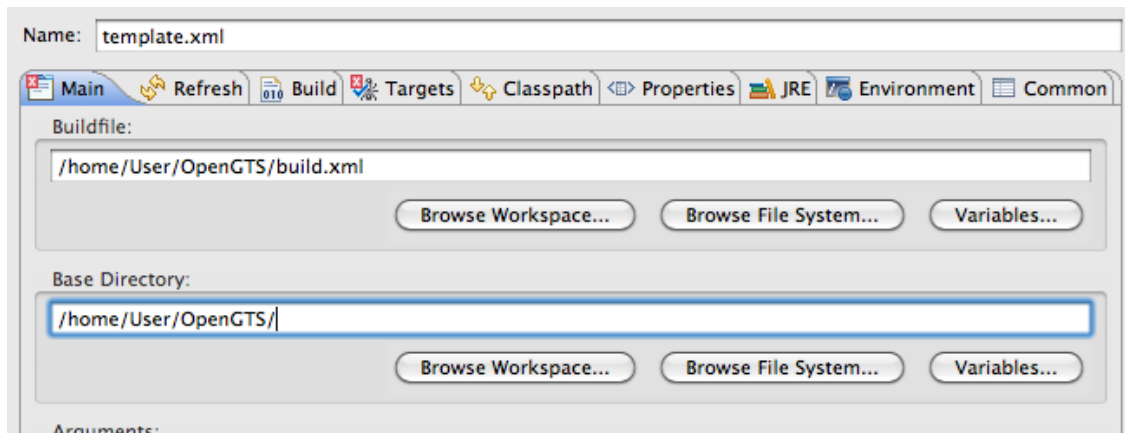
- Add into your *build.xml* or *build_custom.xml* a new target like this:

```
<target name="template.run" depends="template"
  description="Run 'Device Communication Server' template ...">
  <echo message=" template run ..."/>
  <java fork="true" classname="org.opengts.servers.template.Main">
    <arg value="-conf=${Env.GTS_HOME}/default.conf"/>
    <arg value="-start"/>
  </java>
  <classpath>
    <path refid="compile.classpath"/>
    <path location="${build.lib}/template.jar"/>
  </classpath>
</target>
```

- Open **Run -> External Tools -> External Tools Configurations**
- Select *Ant Build* then click on *New launch configuration* button



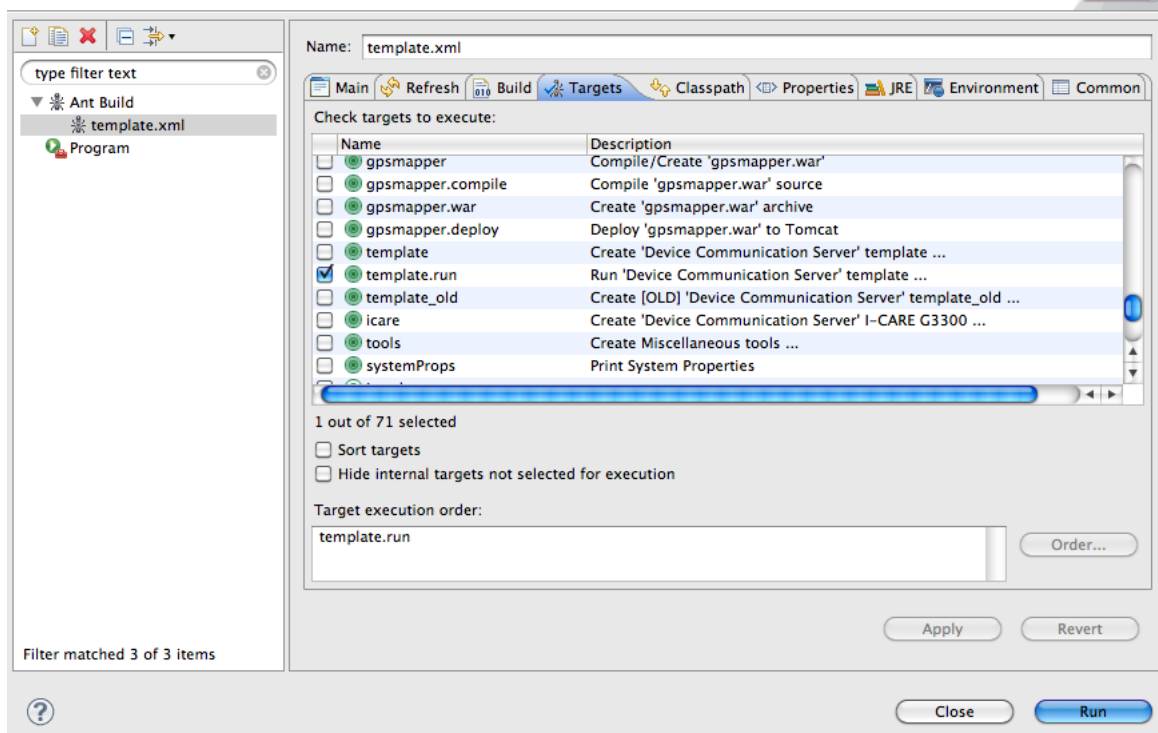
- Add a new „**Ant Build Configuration**“ and set
 - Name: “**template.xml**”
 - Buildfile: “**/home/user/OpenGTS/build.xml**”
 - Base Directory: “**/home/user/OpenGTS**”



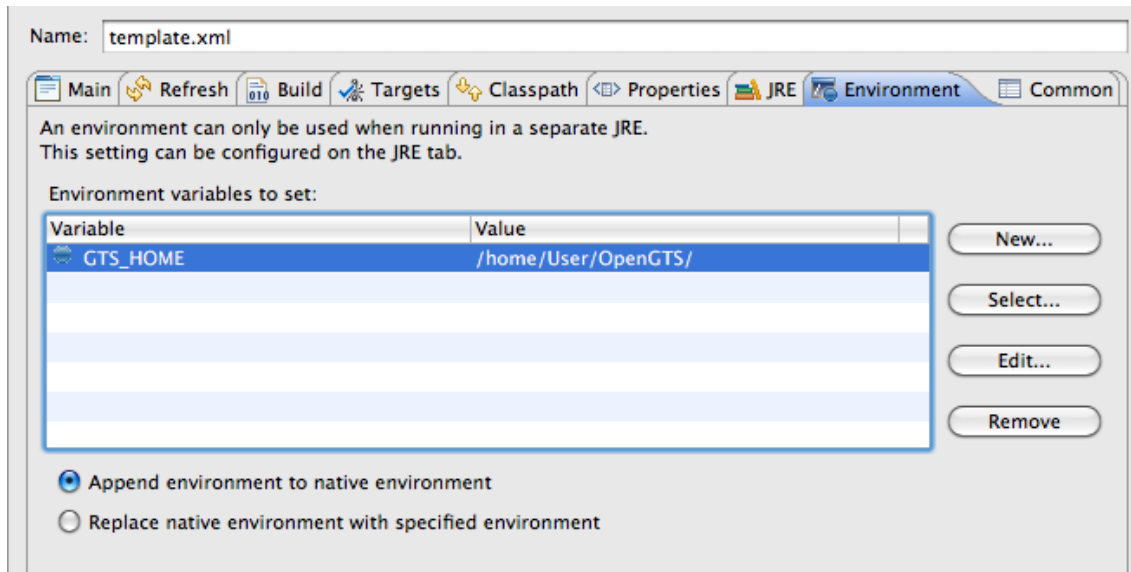
- Select under Tab „**Targets**“ template.run Ant-Target

Create, manage, and run configurations

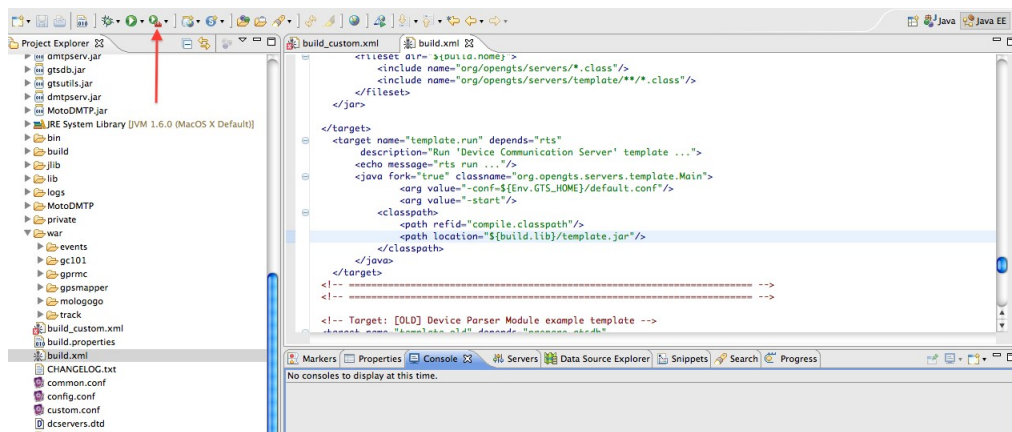
Run an Ant build file.



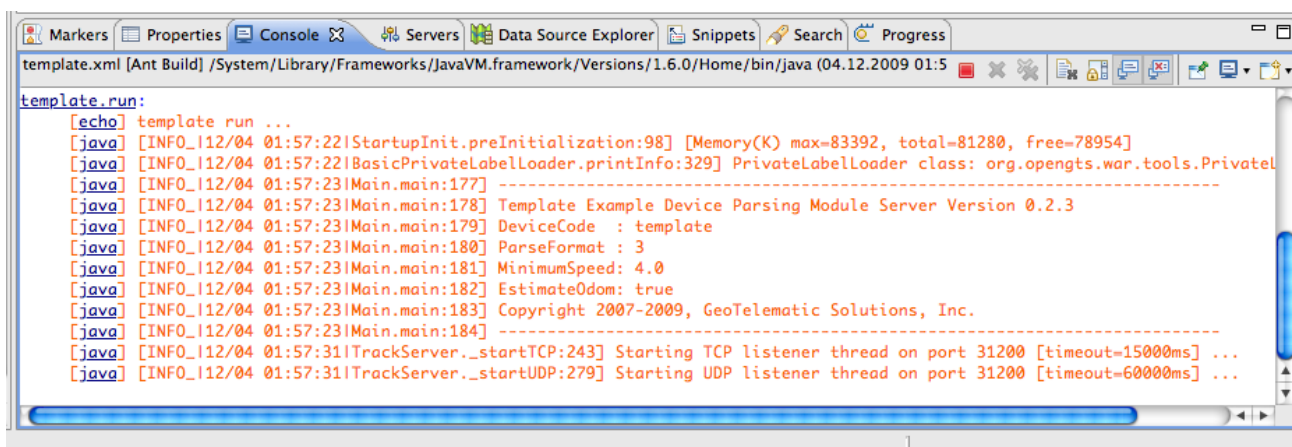
- Set on Tab „Environment“ the GTS_HOME variable like „Base Directory“



- „Apply“ and „Close“
- Now you can Run your template server in eclipse starting from External Tools



- Console window show the output



Debug remote tomcat application using Eclipse

Eclipse can connect to Tomcat and give you the ability to debug your WAR applications. This document describe some basic setup to start tomcat in debug mode and use Eclipse, for more information see tomcat and Eclipse documentation.

- Start tomcat in debug mode:

Default starting script is catalina.sh (or catalina.bat) under Tomcat bin directory.

Use the parameter **jpda** to start tomcat, this run in debug mode and use the port 8000 (default configuration) to give to external debug program all information needed.

Simple run:

```
> ./catalina.sh jpda start
```

or in Windows OS:

```
> catalina.bat jpda start
```

Under Fedora (Release 11) starting script is /usr/bin/dtomcat5, so run:

```
> /usr/bin/dtomcat5 jpda start
```

TIPS: to show all starting options run: catalina.sh -h (or without parameters)

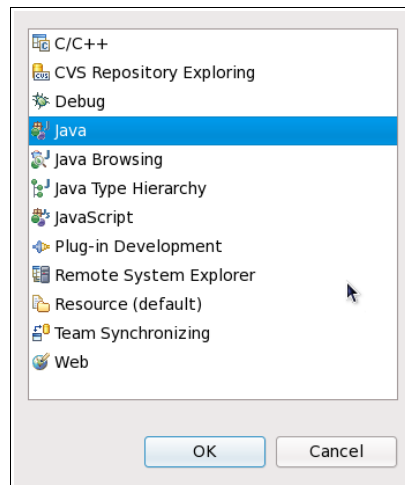
You obtain something like this:

```
Found JAVA_HOME: /usr/lib/java
Please complete your /etc/tomcat5/tomcat5.conf so we won't have to look for it next time
Using CATALINA_BASE:   /usr
Using CATALINA_HOME:   /usr
Using CATALINA_TMPDIR: /usr/temp
Using JRE_HOME:
Usage: dtomcat5 ( commands ... )
commands:
  debug          Start Catalina in a debugger
  debug -security Debug Catalina with a security manager
  jpda start     Start Catalina under JPDA debugger
  run            Start Catalina in the current window
  run -security  Start in the current window with security manager
  start          Start Catalina in a separate window
  start -security Start in a separate window with security manager
  stop           Stop Catalina
  stop -force    Stop Catalina (followed by kill -KILL)
  version        What version of tomcat are you running?
```

- Create debug profile in Eclipse

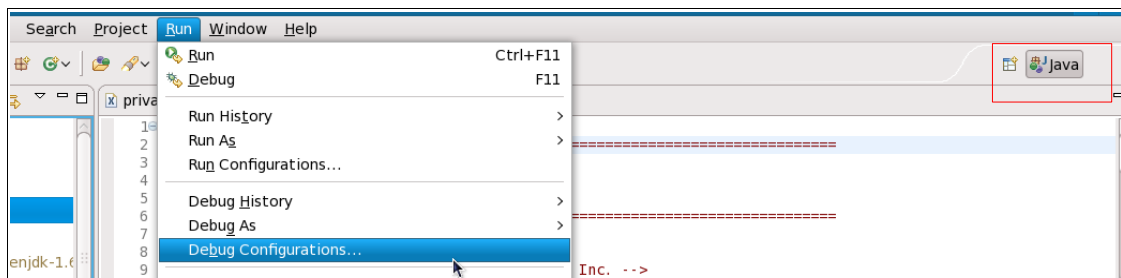
Select the OpenGTS project in the *Navigation pane*

Open the *Java* perspective

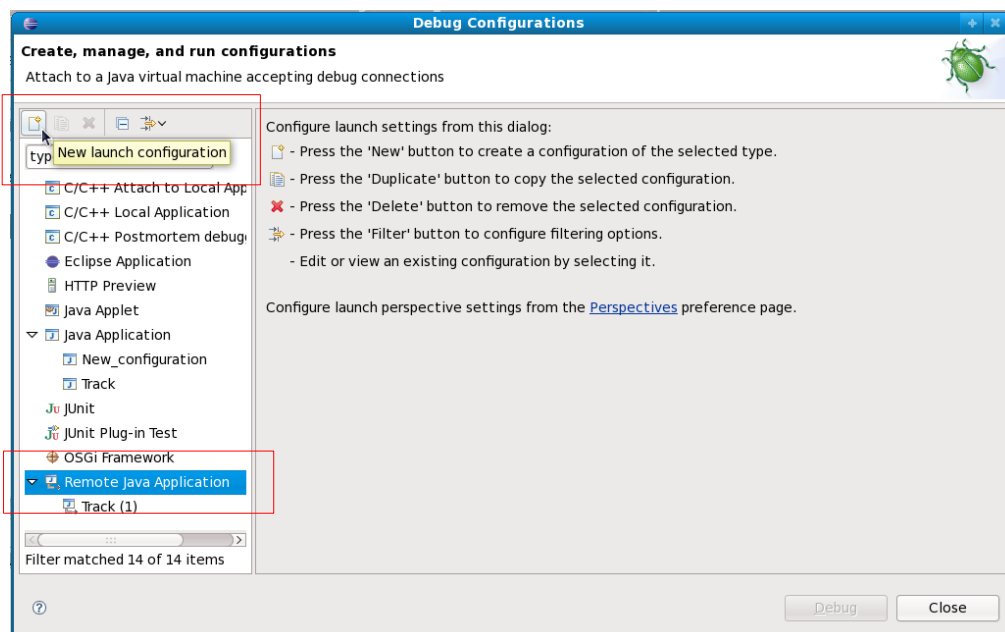


Make sure that Java icon is showed in the upper right corner

Select from menu bar **Run->Debug Configurations**



Select *Remote Java Application* than click on the new button to create a *New launch configuration*

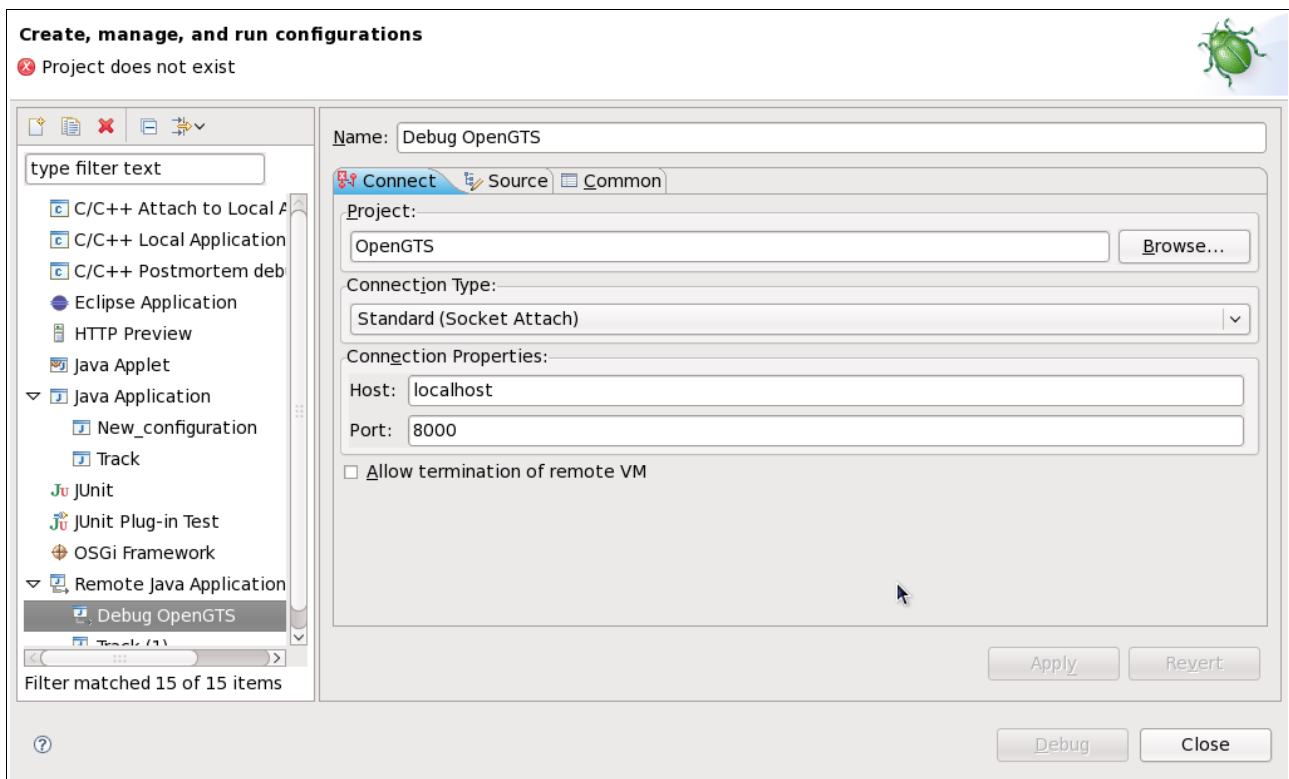


Name: assign the *name* for the debug configuration

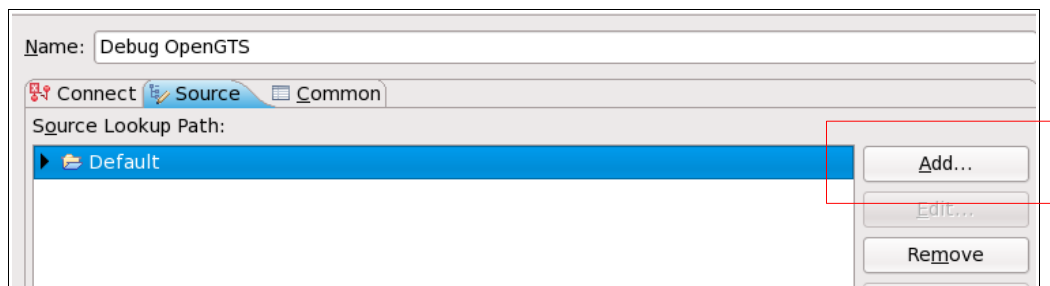
Project: select the OpenGTS project

Host: the host name where tomcat run or *localhost*

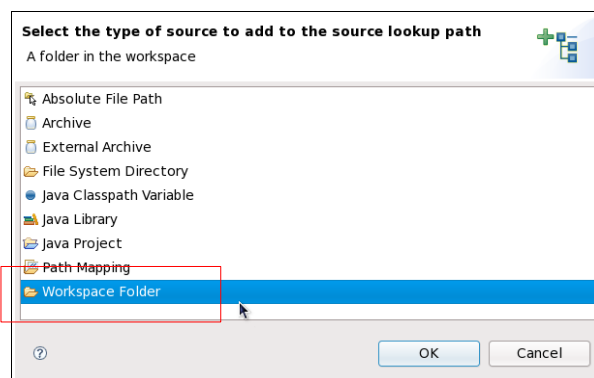
Port: 8000 is the default *jpda* port



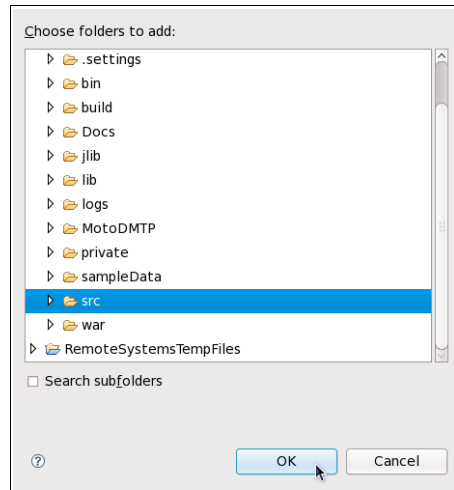
Add a new source path



Select *Workspace folder*



And choice the *src* folder of OpenGTS project



Note: Probabily you must add the war folder too (I have not tested deeply).

Now press F11 on your keyboard or select Run->Debug.

Eclipse ask to you to open the Debug prospective then run you war application on the remote tomcat server and show all debug informations.