RoboCode README

Introduction

Welcome to the De/Coding Openet Event 2019!

So, what is RoboCode? Robocode is a programming game where the goal is to code a robot to battle against other robots in an Arena. Each arena will run for 100 rounds.

Basic Rules

Robots submitted for competition must extend the ***robocode.Robot*** class, **not** the *robocode.JuniorRobot* or *robocode.AdvancedRobot* classes. Winning robots are examined, validated and code reviewed.

Please see below on how to get up and running, by installing all of the necessary software.

Installation

All files necessary are available on the Openet USB sticks.

[Windows Only] Installing 7-Zip

> **BEWARE - Here be dragons!**
>
> **There is a known issue regarding un-zipping the compressed files supplied on the USB key for Windows users - particularly Windows 7 users.**
>
> To get around this issue please perform the following instructions:

1. On the USB key there is a directory called '7-Zip'. Within that directory there is two directories - Windows64 and Windows86.
2. Go into the directory that best describes your Operating System.
3. Execute the .exe installer found within the directory and install 7-Zip.
4. Leave all options to default when installing.

Extraction

Copy everything from the USB onto your local disk.

This guide assumes you have copied the entire contents of the USB stick onto a chosen directory and continues to refer to this local directory.

The directory structure is as follows:

```
7-Zip
   license.txt
   Windows64
      7z1604-x64.exe
   Windows86
       7z1604.exe
Eclipse-PreConfigured
   Eclipse-Workspace
      OpenetFastStart
   Linux32.zip
   Linux64.zip
   Mac64.zip
   Windows64.zip
   Windows86.zip
Eclipse-PreConfigured.zip
JDK
   Linux64.zip
   Linux86.zip
   Mac.zip
   Windows64.zip
   Windows86.zip
LICENSE.txt
README.html
README.pdf
RoboCode-Basics.html
RoboCode-Basics.pdf
robocode-executable
   robocode-1.9.3.0-OpenetEdition-setup.jar
   source
       robocode-1.9.3.0-OpenetEdition-src.zip
test-robot
    openet.OpenetTestBot_1.0.jar
    supersample.SuperCorners_1.0.jar
    supersample.SuperTracker_1.0.jar
    supersample.SuperTrackFire_1.0.jar
```

Installing the JDK

Available JDKs

JDK Zips

In the JDK directory you will find the following zip files:

- Windows86.zip
- Windows64.zip
- Mac.zip
- Linux64.zip
- Linux86.zip

Extract the one that best describes your Operating System.

Windows

1. Depending on your OS architecture, either extract **JDK -> Windows64.zip** or **JDK -> Windows86.zip**
2. After extraction go into the extracted directory - either Windows64 or Windows86
3. Execute the JDK installation exe and follow the onscreen instructions
4. After the installation process you will need to add Java to your ENV PATH
    1. Right Click on MyComputer and click on properties .
    2. Click on Advanced tab
    3. Click on Environment Variables\
    4. Create a new class path for JAVA_HOME
    5. Enter the Variable name as JAVA_HOME and the value to your jdk bin path ie **c:\Programfiles\Java\jdk-1.8** and
    6. Modify the PATH 'System Variable', attach the following to the end of the 'System Value' (excluding quotation marks)

        ```
        ;%JAVA_HOME%/bin
        ```

Test Java is available on your ENV PATH:

1. Click Start and type "command prompt" in the search box
2. In the command prompt, type:

```
java -version
```

You should get back the version of Java - it should be **1.8.xxx**

Linux

1. Depending on your OS architecture, either extract **JDK -> Linux64.zip** or **JDK -> Linux86.zip**
2. After extraction, to install the JDK on Linux simply untar the Linux JDK .tar file into any directory (usually opt)
3. Set an environmental variable JAVA_HOME to the bin directory of the extracted JDK in your .bashrc file in your /home/<username> directory (see below example)
4. Add JAVA_HOME to your PATH also in your .bashrc file as per the example below

```
JAVA_HOME=/opt/java/jdk1.8.0_60
PATH=$PATH:$JAVA_HOME/bin
export PATH
```

Test Java is available in your ENV

1. Open a terminal
2. In the terminal, type:

```
java -version
```

You should get back the version of Java - it should be **1.8.xxx**

Mac

1. Extract the Mac JDK zip.
2. Installing JDK on Mac requires executing the .dmg file found in the Mac subdirectory of JDK

Test Java is available in your ENV

1. Open a terminal
2. In the terminal, type:

```
java -version
```

You should get back the version of Java - it should be **1.8.xxx**

Installing RoboCode

To install RoboCode, please see below for your OS specific installation instructions:

Windows

1. Go to **robocode-executable**
2. Double click on **robocode-1.9.3.0-OpenetModification-setup.jar**
3. You will be asked where to store the game files - simply leave it as default (Openet-Robocode-Competition)
4. Click Yes
5. You will then receive a pop-up saying RoboCode has installed.

Linux

1. Open a terminal
2. Go to **robocode-executable**
3. Execute: **java -jar robocode-1.9.3.0-OpenetModification-setup.jar**
4. You will be asked where to store the game files - simply leave it as default (Openet-Robocode-Competition)
5. Click Yes
6. You will then receive a pop-up saying RoboCode has installed.

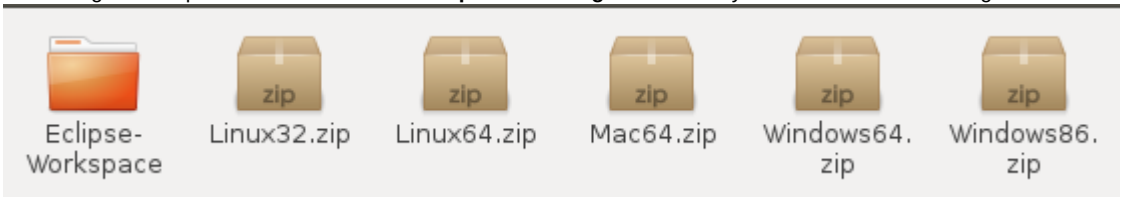
Mac

1. Go to **robocode-executable**
2. Double click on **robocode-1.9.3.0-OpenetModification-setup.jar**
3. You will be asked where to store the game files - simply leave it as default (Openet-Robocode-Competition)
4. Click Yes
5. You will then receive a pop-up saying RoboCode has installed.

Installing Presetup Eclipse

> **NOTE**
>
> **This step is highly recommended to getting you up to speed with development.**
>
> This installs Eclipse Neon IDE preconfigured with RoboCode and a base workspace

1. Make sure the JDK is installed - see steps above
2. Extract the **Eclipse-PreConfigured zip** into your local directory. Ensure it gets extracted into the 'Eclipse-PreConfigured' directory - it's just easier to manage and organise.
3. Pre-Configured Eclipse instances are in the '**Eclipse-PreConfigured**' directory. You will see the following:

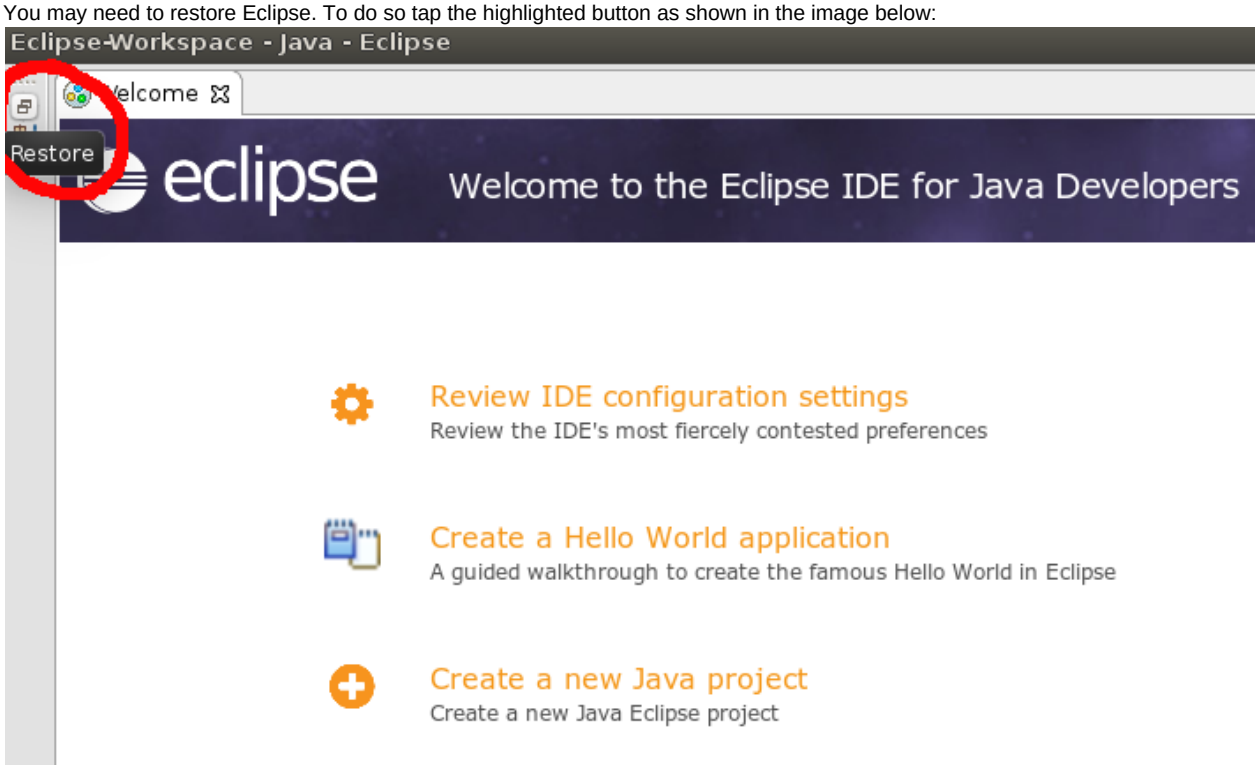


4.
5. Extract the Zip which best describes your OS.
6. Go into the extracted directory
7. You will then find an executable for '**eclipse**'.
8. If you're on Windows it will be **eclipse.exe**
9. If you are on Linux the executable will simply be '**eclipse**'
10. On Mac the executable will be **Eclipse.App**.

11. Start Eclipse

> **Running Eclipse on Ubuntu**
>
> If you are on Ubuntu (or a deviation) you may need to execute Eclipse within a terminal. To do so, drop into a Terminal, and execute './eclipse'.

- Eclipse will ask you for a Workspace location
- Select the Eclipse-Workspace found within the Eclipse-PreConfigured directory
- You may need to restore Eclipse. To do so tap the highlighted button as shown in the image below:



- Open the OpenetFastStart project.
- Test the libraries and JavaDoc work correctly by opening the **OpenetFastStart.java** class - make sure there are no errors and that you can hover over a Class and view its JavaDocs.

> **NOTE**
>
> **Rename the Java class to your team name.**
>
> Rename "**OpenetFastStart**.java" to "**YourTeamName**.java", replacing "**YourTeamName**" with the team name you registered earlier on.
>
> It must match your team name exactly. If it does not match your team name at time of submission, it may result in your robot being skipped for that round.

Running RoboCode

- Go to the directory where you installed RoboCode* and find the executable called '**robocode**'.
  - Windows - **C:\Openet-Robocode-Competition**
  - Linux / Mac - **/home/<user>/Openet-Robocode-Competition**
- Depending on the operating system, this will either be a :
  - **.exe** for Windows or
  - .**sh** for Linux / Mac.

Mac / Linux

For Mac / Linux users - you may need to go into a terminal to execute RoboCode. Execute the following in a terminal :

```
$ cd ~/Openet-Robocode-Competition

$ ./robocode.sh
```

After you execute RoboCode it will start up, present a splash screen and then shows the actual application.

NOTE

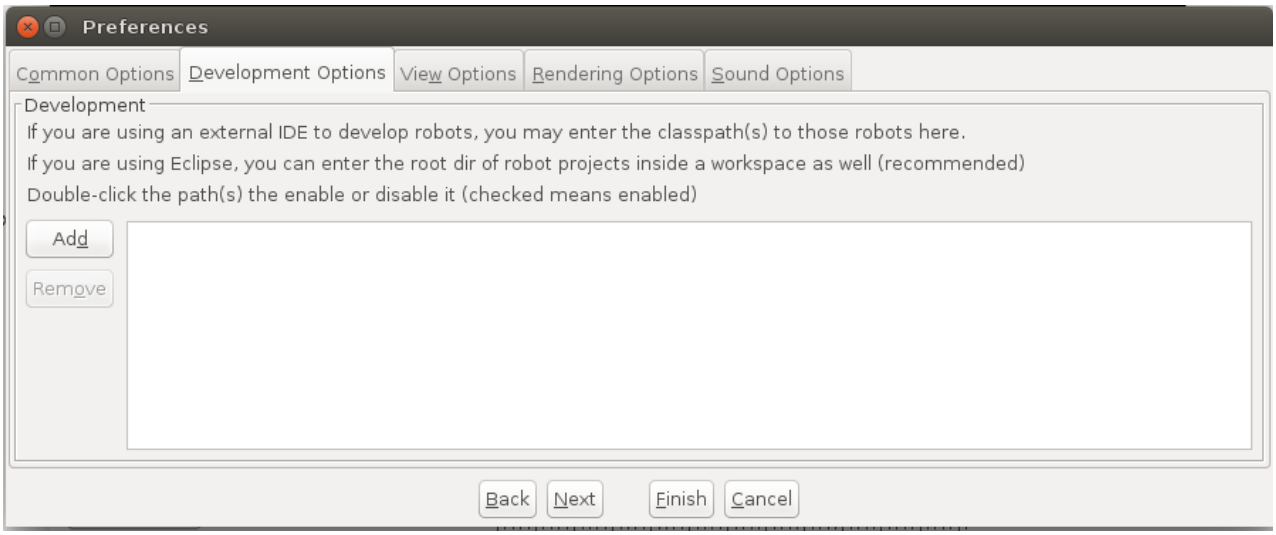
Sometimes when RoboCode loads up fully there may be a Previous Battle screen, just click O.K.. It happens now and again.

Importing Eclipse Project

NOTE

Only perform this step if you have performed the previous Eclipse step - 'Installing Presetup Eclipse'

- After running the Install Eclipse steps above, then you can import it into RoboCode.
- In RoboCode, go to Options -> Preferences -> Development Options tab.



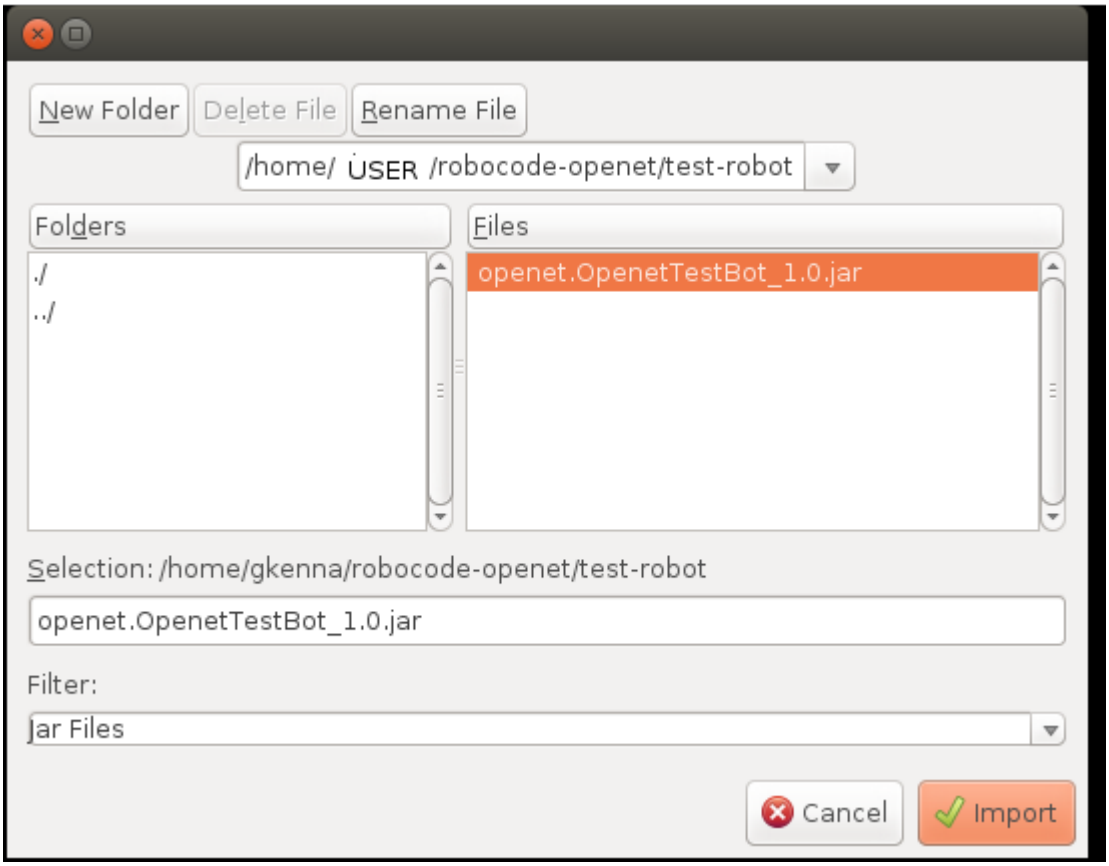
- Click 'Add'. You are presented with a screen.
- Find the directory in which your bot source code lives, it should be something like :
    - Windows - **'C:\home\CopiedUsb\Eclipse-PreConfigured\Eclipse-Workspace\OpenetFastStart'**
    - Linux / Mac - **'~/CopiedUsb/Eclipse-PreConfigured/Eclipse-Workspace/OpenetFastStart'**
- Click Ok. Your Eclipse project is added.
- Click Finish.

Any changes you make to your Robot in Eclipse now automatically updates on RoboCode - it compiles automatically.

Importing Test Bot

Now that it's up and running we are going to Import the Openet Test Bot, a robot you can use to fight against with your own bot.

1. In RoboCode Click Robot -> Import Robot or Team.
2. A dialog appears.
3. Go to the directory '**test-robot**' - it is the directory you copied over from the USB.
4. Add the **OpenetTestBot_1.0.jar** robot.



If it is successful you see a "Bot added successfully" message.

You can also choose one of the more advanced test bots, such as:

- **supersample.SuperCorners_1.0.jar**
- **supersample.SuperTracker_1.0.jar**
- **supersample.SuperTrackFire_1.0.jar**

Test Battle

Now that we imported the test bot let's put them to battle.

- Go to Battle -> New. You sees the *openet* package has two Bots - **YourBot** and **OpenetTestBot**.
- The **FastStart** bot has its source code available.
- Add the two bots
- Click Start Battle.
- The two robots start fighting to the death!

Improving Your Bot

After the test battle has finished, we will now improve your bot.

We firstly access the API's and then open up the Source Editor.

API Guide

The API Guide will become your best friend. To access it:

- Click **Help -> RoboCode Api.**
- This opens up a new Browser with an offline version of the JavaDocs

Extending Supplied Robot

The robot will be automatically compiled and imported into RoboCode each time the source code is saved.

- Modify the Robot colour by changing the code:

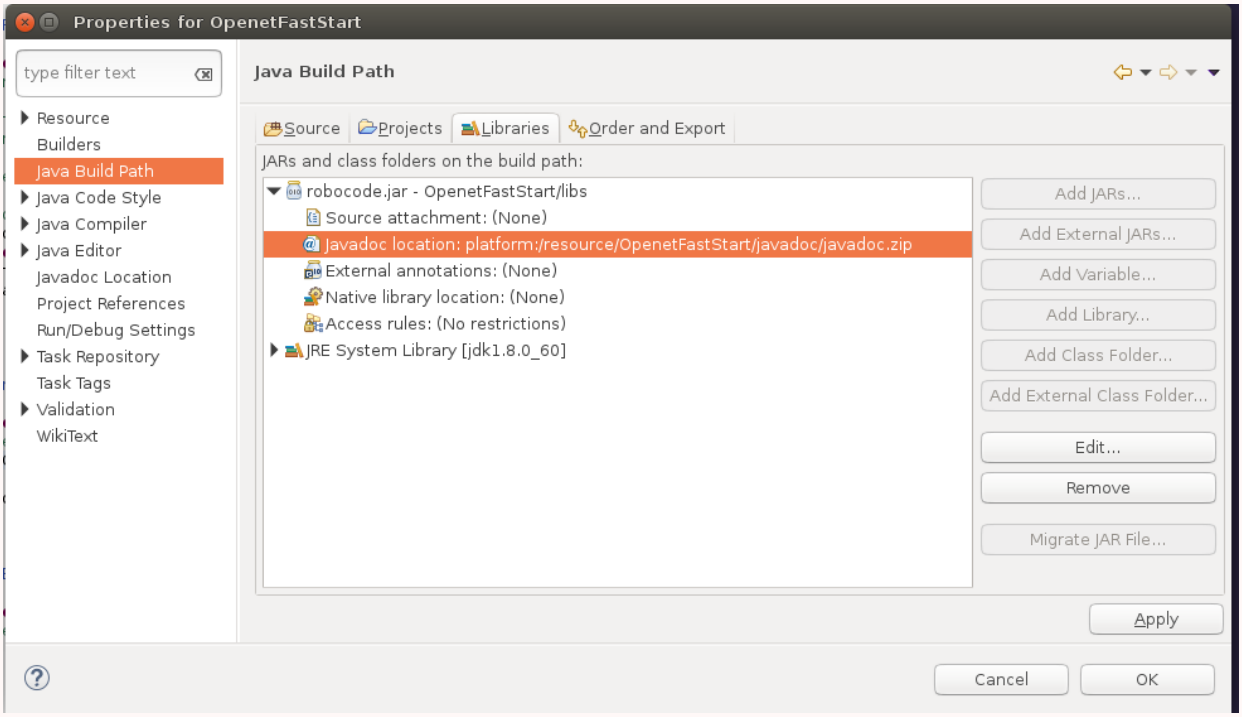
```
setColors(Color.red,Color.blue,Color.green); // body,gun,radar
```

- Save the code
- Go back into Robocode
- Add your bot to the battle
- You should see the bot's colour change.

Don't modify the actual Eclipse project name - This will remove Javadocs.

If you do happen to modify the Eclipse project name then to remedy Javadocs do the following:

- Right click the Eclipse Project -> Properties
- Click the tab Java Build Path
- Look for the Robocode.jar lib and expand it
- Select Javadocs and select 'Edit'
- Point towards the Javadocs.zip location.
- If you're reading this, inform your Arena Master and say 'Openet MicroServices'. It may be worth your while.



Packaging

The final step is packaging - this step involves getting your bot into a .jar format and giving it to the Arena Master.

- Go to the main RoboCode application.
- Click **Robot -> Package Robot**.
- You will now be presented a screen to package up your robot.
- Select your Robot
- Click next.
- Make sure you 'Include The Source Files" and "Include Data Files".
- Select a version number for your bot, a description and the team name.
- Click package.
- This will then package your bot into a jar.

NOTE

PLEASE TAKE NOTE OF WHERE THE ROBOT IS PACKAGE - YOU WILL NEED IT FOR THE NEXT STEP

Submitting Your Robots

- Put your robots jar on the usbs provided.
- Pass it on to the arena crew for the arena master to include it in the match.