Building and Testing the TorXakis Eclipse Plug-In

# Building the Plug-In

To build the TorXakis Eclipse Plug-in do the following:

1. Start Eclipse.
2. If you didn’t create a “development” Eclipse workspace yet, this can be done in two different ways:
   1. When Eclipse starts, the “Workspace Launcher” dialog opens. In the “Workspace” field select the location where you want to store the “development” workspace (make sure the location ends with “\development” and click “OK”.
   2. After Eclipse has started, select “File 🡪 Switch Workspace 🡪 Other…” and the “Workspace Launcher” dialog opens. From this point on follow the steps mentioned in step a.

If you already have a development workspace, make sure it is selected.

1. If you created a new “development” workspace in step 2, execute the following steps. If not, continue with the next step.
   1. From the Eclipse “File” menu select “Import…”. The “Import” dialog opens.
   2. Select “General 🡪 Existing Projects into Workspace” and click “Next”.
   3. Click the “Browse…” button and in the “Browse For Folder” dialog select the folder containing <https://github.com/TorXakis/SupportEclipse>

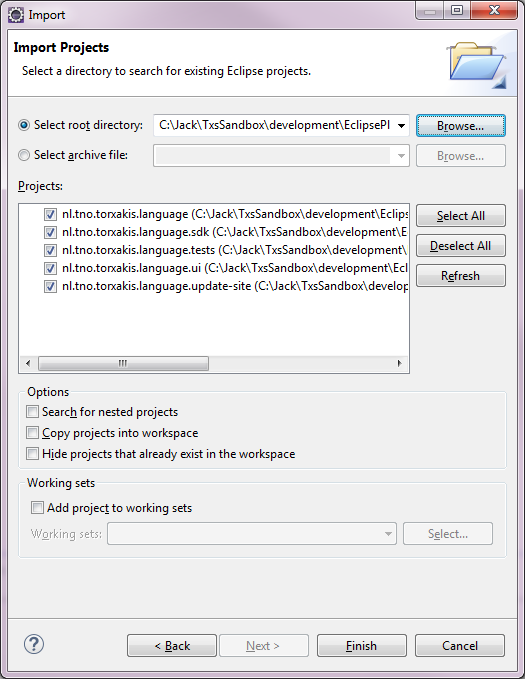


Figure 1: Importing the EclipsePlugin projects

* 1. The “Import” Dialog should contain all of the projects as shown in Figure 1. Click “Finish”.

The required plug-in projects are now loaded in the workspace. Note that are still some errors in the projects (red box with white cross icons in the Project Explorer). This is correct.

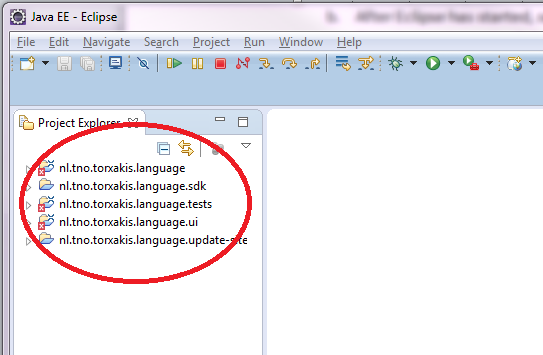


Figure 2: Some projects contain errors which is OK

1. In the “Project Explorer” navigate to the folder “org.torxakis 🡪 src 🡪 org.torxakis”.
2. Right-click on the file “GenerateTorXakisDsl.mwe2” and select “Run As 🡪 MWE2 Workflow”. The “Errors in Workspace” dialog might appear. Ignore this and click “Proceed”.
3. In the “Console” tab the following message in red might appear:

\*ATTENTION\*

It is recommended to use the ANTLR 3 parser generator (BSD licence - http://www.antlr.org/license.html).

Do you agree to download it (size 1MB) from 'http://download.itemis.com/antlr-generator-3.2.0-patch.jar'? (type 'y' or 'n' and hit enter)

If so, click in the “Console” window and enter ‘y’ and press “Enter” to agree to download the ANTLR 3 parser.

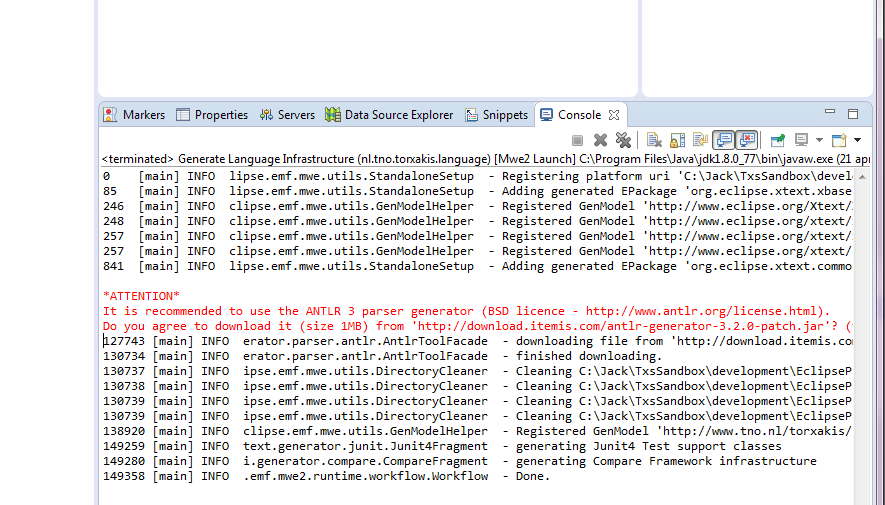


Figure 3: The ANTLR 3 parser generator is needed

1. When the workflow has run to completion, the “Console” window will show “Done.”. Now, if there were errors in projects, all errors should be gone (no red icons in project Explorer. This might take some time to update).
2. If org.torxakis.ui.tests package still shows error icon, then you may not have “Eclipse Plug-in Development Environment” installed. From Eclipse “Help” menu select “Install New Software…”. In the “Install” dialog, select repository of your Eclipse version (Kepler, Neon, Oxygen, etc). Type “Eclipse Plug-in Development Environment” to filter box; select “Eclipse Plug-in Development Environment” under “General Purpose Tools”. Click Next (2 times), accept license agreement, and Finish; restart Eclipse when asked. The error mark on org.torxakis.ui.tests package should be gone.

# Testing the Plug-In

To test the plug-in, follow the following steps:

1. In the “Project Explorer”, right-click on the “org.torxakis.ui” project and select “Run As 🡪 Run Configurations…”. The “Run Configurations” dialog opens.

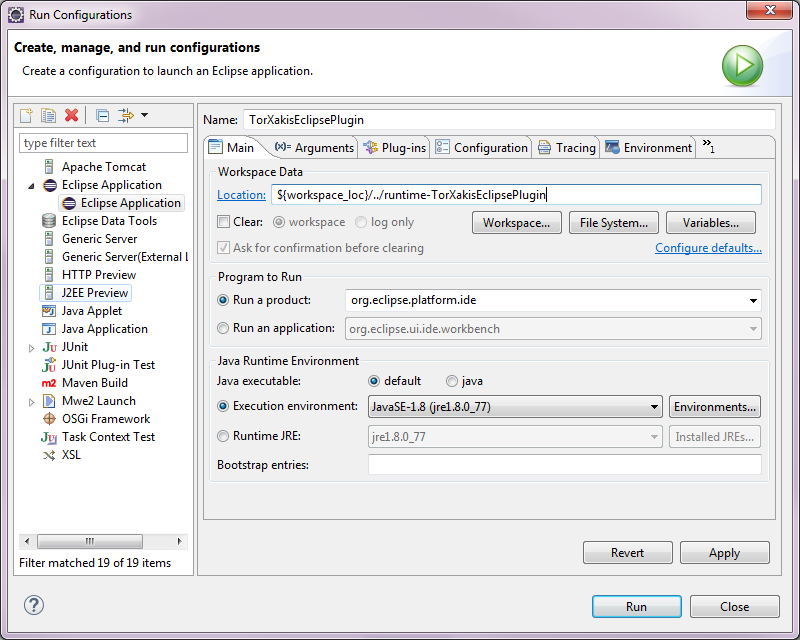


Figure 4: Creating a run configuration for the Eclipse application

1. In the “Name” field enter “TorXakisEclipsePlugin”.
2. In the “Workspace Data” enter “${workspace\_loc}/../runtime-TorXakisEclipsePlugin”.
3. Click “Apply” and then click “Run”.

The Eclipse runtime environment is started. You now have two instances of Eclipse running. The next steps need to be carried out in the runtime environment (name in title bar: “Resource – Eclipse Platform”).

To test the plug-in do the following:

1. From the Eclipse “File” menu select “Import…”. The “Import” dialog appears.
2. Select “General 🡪 Existing Projects into Workspace” and click “Next”.
3. Click “Browse…”, the “Browse For Folder” dialog opens.
4. Select the folder test contained in the folder containing <https://github.com/TorXakis/SupportEclipse> and click “Finish”.

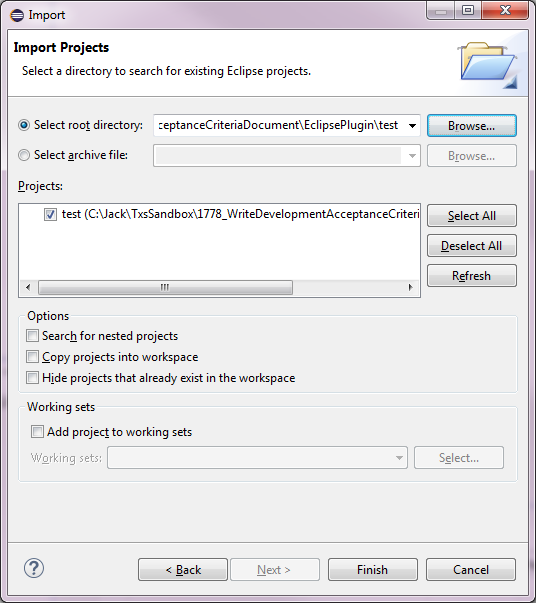


Figure 5: Importing the test project in the runtime Eclipse

1. The “test” project is now visible in the “Project Explorer”. Open the test folder and select all “.txs” files in it.

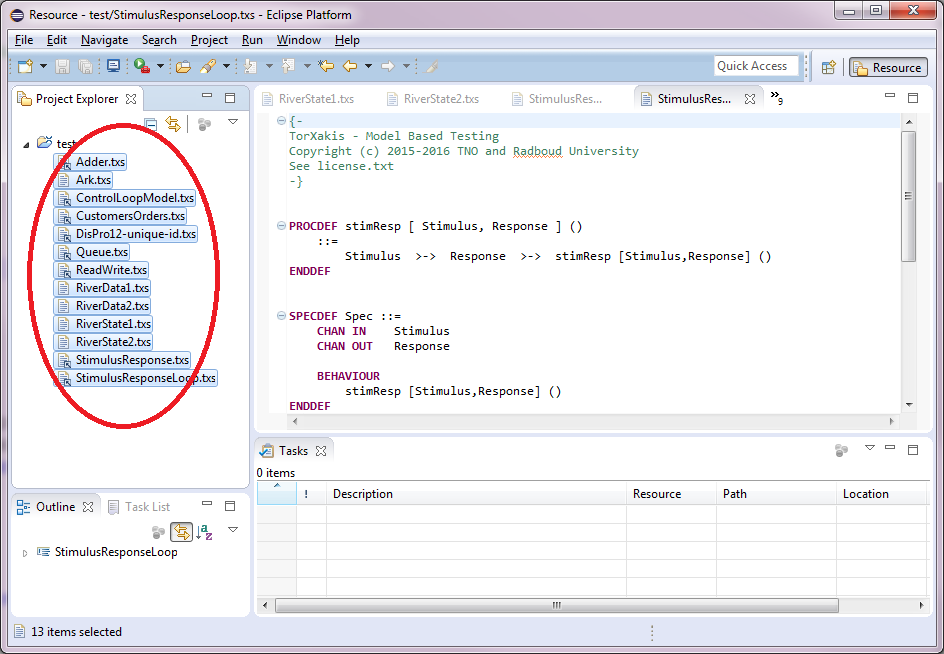


Figure 6: Selecting all TorXakis specifications

1. Right-click one of the selected files and select open. This will open editors for all .txs files.
2. Now check that there are no errors in the file. If a red rectangle appears in front of the file name, there is an error in the plug-in. Consult the developer who made changes to the Eclipse Plug-in whether he has introduced an error.
3. When finished close the Eclipse runtime environment.

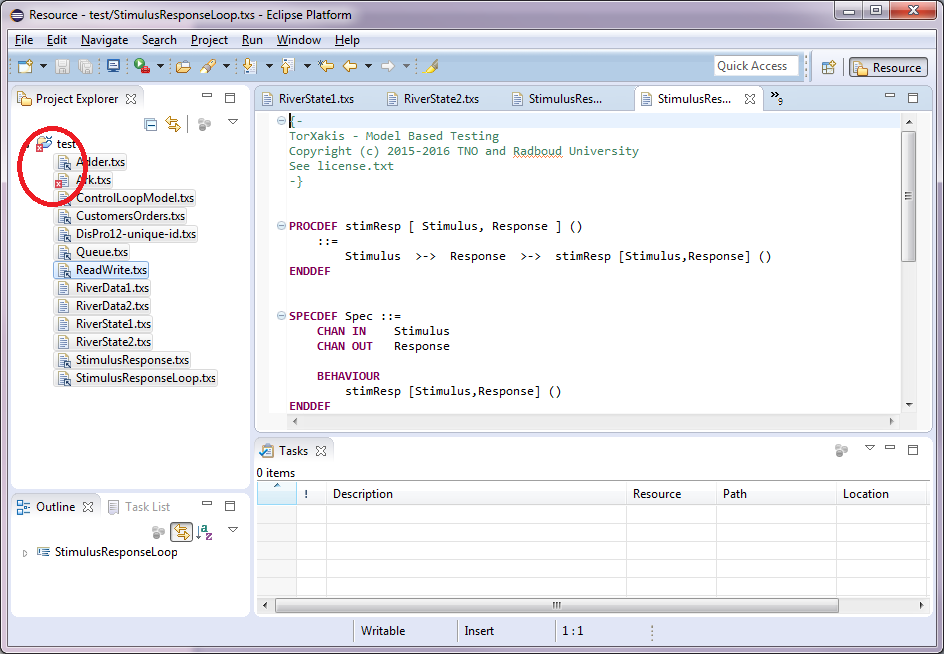


Figure 7: There is an error in one of the specifcations