Eclipse workspace for JSimMAST

César Cuevas Cuesta cuevasce@unican.es

José M. Drake Moyano drakej@unican.es

Patricia López Martínez lopezpa@unican.es

CTR -Computers and Real-Time Group

Electronics and Computers Department

University of Cantabria (Spain)

Eclipse workspace for JSimMAST

Contents

1 Getting your Eclipse workspace with JSimMAST ready to use.

1.1 Download

As indicated in the JSimMAST web page, for the download of a working Eclipse workspace with JSimMAST, you must right click on the indicated link and select "*Save link as*". A ZIP archive named **eclipse_workspace_JSimMAST.zip** will be downloaded.

1.2 Installation

Move the eclipse_workspace_JSimMAST.zip archive to your usual location for Eclipse workspaces (e.g. C:\Workspaces\ ) and extract it using your favourite application (e.g. WinRAR). It will lye the folder **eclipse_workspace_JSimMAST** (Figure 1). Its name can be changed if desired.

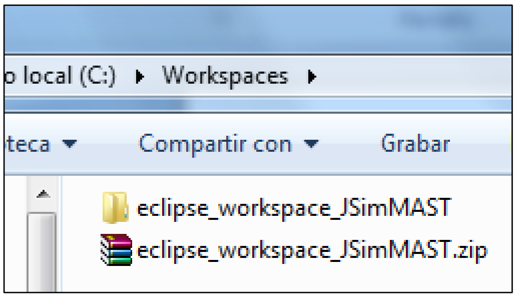
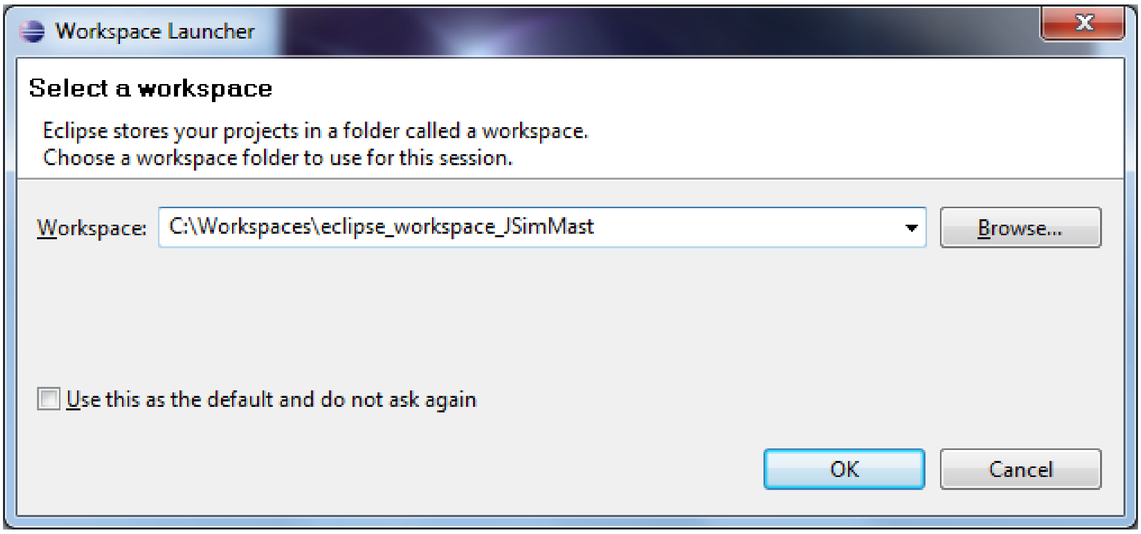

o local (C:) ▸ Workspaces ▸
teca ▾
Compartir con ▾
Grabar
eclipse_workspace_JSimMAST
eclipse_workspace_JSimMAST.zip

Figure 1. The Eclipse workspace for JSimMAST installed

1.3 Launching

Fire your copy of Eclipse and through the Workspace Launcher (Figure 2) select the workspace you have just downloaded, writing down the location in which you have installed (extracted) it or directly browsing to it.



Workspace Launcher

Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\Workspaces\eclipse_workspace_JSimMast

□ Use this as the default and do not ask again

Figure 2. Workspace selection at Eclipse opening.

2 Workspace contents

Once the workspace is open, its initial state presents three projects as contents (Figure 3). The first one is a Java project (JSimMAST) and the other two (JSimMAST_Schemas and SimpleScada) have no special nature.

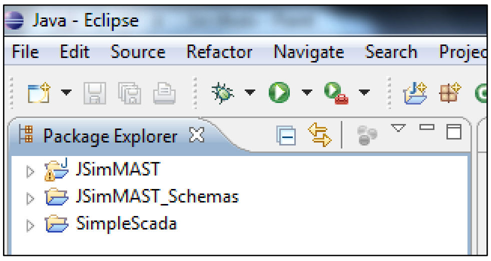


Figure 3 Contents of eclipse_workspace_JSimMAST

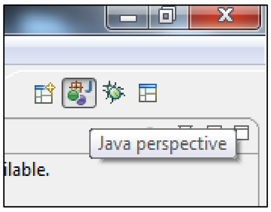


Figure 4. Switching to the Java perspective

Since the main JSimMAST project has Java nature, it is recommended to work on the basis of the Java perspective. Switch to it through the appropriate button (Figure 4) if you are working in a different one

2.1 The JSimMAST project

The JSimMAST project (Figure 5) is a Java project that contains the source code of the simulator. For the user, the most relevant class, *JSimMastGUI*, is within the *guis* package and it is the class representing the GUI of the JSimMAST tool, so it contains a *main()* method and through it the simulator can be launched. Section 3 deals with this issue.

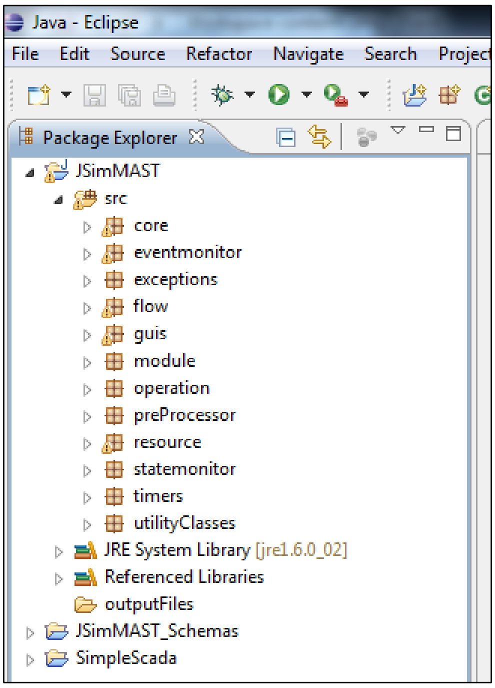


Figure 5. The JSimMAST project showing the source code packages

Problems with the JSimMAST project when launching the workspace for the first time?

If when opening the downloaded workspace for the first time, the JSimMAST project presents a problem marker like the one shown in Figure 6, it is possibly doubt to an issue regarding its java class path.

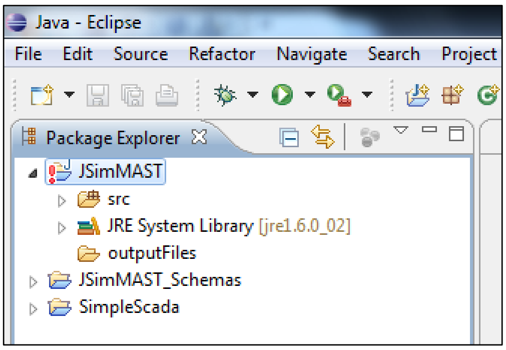

Java - Eclipse
File Edit Source Refactor Navigate Search Project
Package Explorer
JSimMAST
src
JRE System Library [jre1.6.0_02]
outputFiles
JSimMAST_Schemas
SimpleScada

Figure 6. Problem marker

The project makes use of the SWT library for the GUI construction. This library is wrapped into the ***org.eclipse.swt.win32.win32.x86_???????.jar*** *archive, which is located in the plugins folder of your Eclipse installation.*

The exact value of the "????????" token depends on the Eclipse release. For instance, JSimMAST has been developed with Eclipse Helios SR2, for which the SWT library appears in the plugins folder wrapped in the ***org.eclipse.swt.win32.win32.x86_3.6.2.v3659c.jar*** *archive. By default, the java class path for the JSimMAST project references to this JAR archive assuming it is located at* ***D:\Eclipse distros\Eclipse Helios SR2 (3.6.2)\eclipse-java\plugins****. Since this is probably not your case, you must configure the java class path of the JSimMAST project with you own settings, following these steps:*

i. *Right click on the JSimMAST project and select "Configure build path" (Figure 7)*

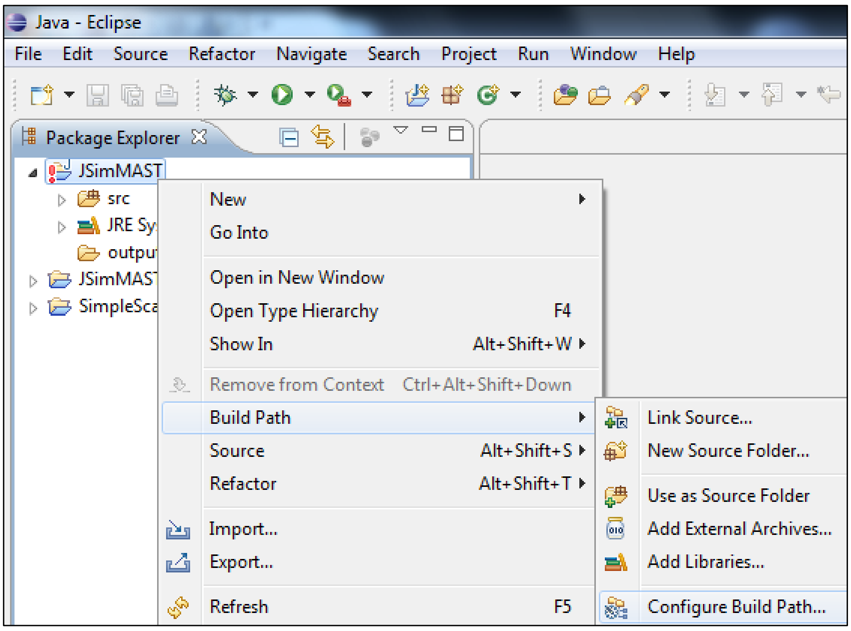
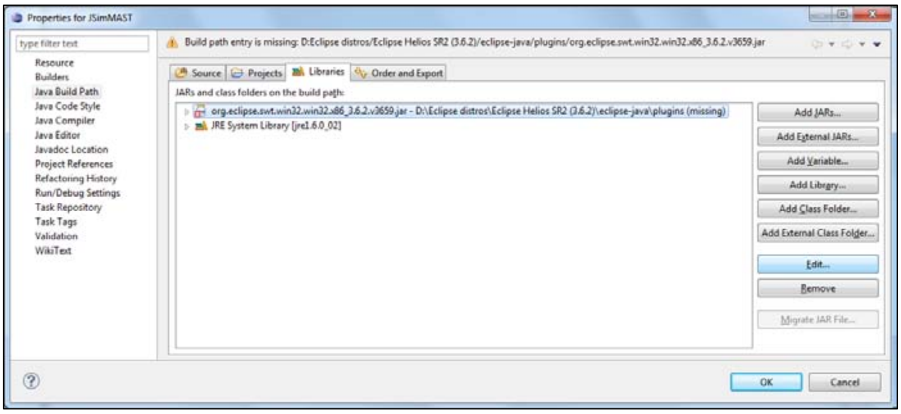


Figure 7. Configure Build Path pop-up option

ii. Select the SWT archive (which appears as missing, as in Figure 8) and click on the Edit button. This will open a file dialog window allowing you to browse your own SWT archive.



Build path entry is missing: D:Eclipse distros/Eclipse Helios SR2 (3.6.2)/eclipse-java/plugins/org.eclipse.swt.win32.win32.x86_3.6.2.v3659.jar

JARs and class folders on the build path:

- org.eclipse.swt.win32.win32.x86_3.6.2.v3659.jar - D:\Eclipse distros\Eclipse Helios SR2 (3.6.2)\eclipse-java\plugins (missing)
- JRE System Library [jre1.6.0_02]

Figure 8. Selecting location of SWT library

2.2 The JSimMAST_Schemas project

This project with no special nature is only meant for containing the W3C-schemas that the XML files involved with the JSimMAST operation must be according to, namely (see Figure 9):

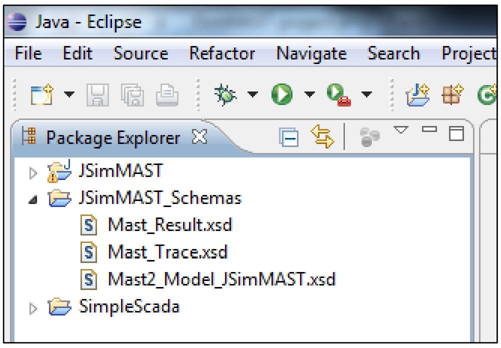


Figure 9. The JSimMAST_Schemas project

- Mast2_Model_JSimMAST.xsd, for the model files.
- Mast_Result.xsd, for the results files generated by JSimMAST.
- Mast_Trace.xsd, for the traces files generated by JSimMAST.

Working with the schemas located here is more than convenient, although not mandatory. The XML file of the provided example (see section 2.3) specifies its *xsi:schemaLocation* attribute expecting that the schema is inside this project and at the same level of the folder (project) containing the XML file. i.e.

xsi:schemaLocation="http://mast.unican.es/xmlmast/model **../JSimMast_Schemas/ Mast2_Model_JSimMAST.xsd**"

However, you can obviously place the schema wherever you want and edit your XML model files accordingly. Different is the point with the results and traces files generated by JSimMAST. Although you can enter the path where they must be generated (see the JSimMAST user guide), they will be generated with a *xsi:schemaLocation* attribute with same value as above, so, depending on the path you specified for the generation, maybe the xsi:schemaLocation attribute of the generated files should be modified.

2.3 The SimpleScada project

It is recommended, although not at all mandatory, to create a specific project with no special nature for every model (or models family) to simulate and to save there the results and traces files generated by JSimMAST when running a simulation on that model. This way, there would be no problems regarding the relative locations between XML files and W3C-Schemas. In addition, it is very friendly to enter the models location in the GUI as a path relative to the workspace root, starting with the string "../".

As an example of this organization strategy – which remember, is only a proposal of how to store your models to be simulated, being the final decision entirely up to you – the project with no special nature SimpleScada is presented within the workspace. As shown in Figure 10, it contains the XML file for the model of the *SimpleScada* application.

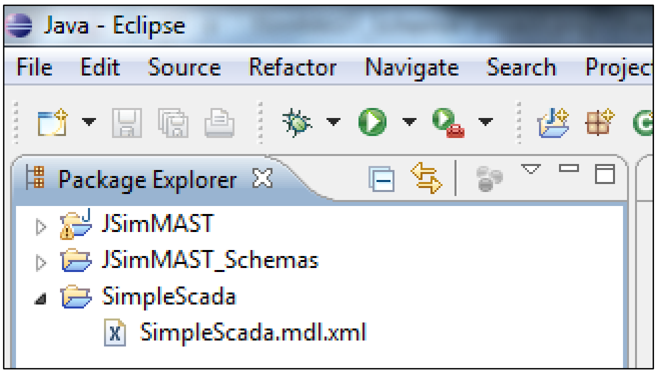


Figure 5. SimpleScada project

Working with this example, the path that should be provide to the GUI in the model text field would simply be **../SimpleScada/SimpleScada.mdl.xml**, and analogue ones for the other files.

3 Launching JSimMAST

3.1 Running JSimMastGUI.

A first option is to launch JSimMAST directly from the *JSimMastGUI* class, since it is the main class of the application (the one with the *main()* method). Right click on it and select "Run as Java application..." (Figure 11).

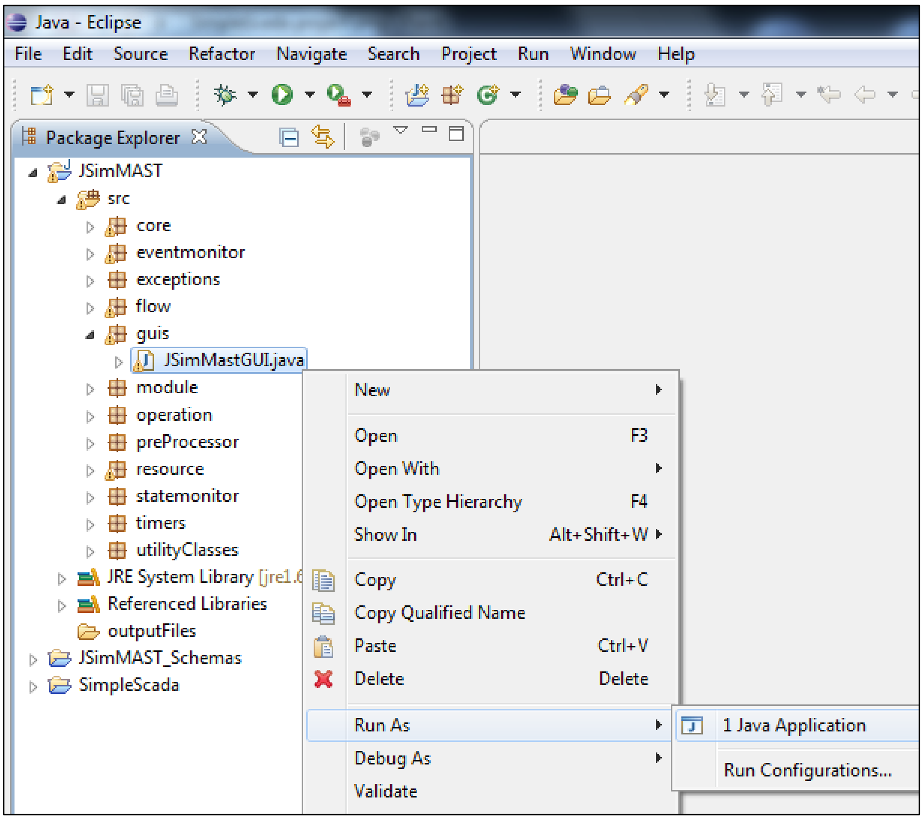


Figure 6. Running JSimMastGUI as a Java application

3.2 Using a launch configuration.

The workspace also provides a shortcut for launching JSimMAST. It consists of a launch configuration ready to be invoked from the toolbar (Figure 12).

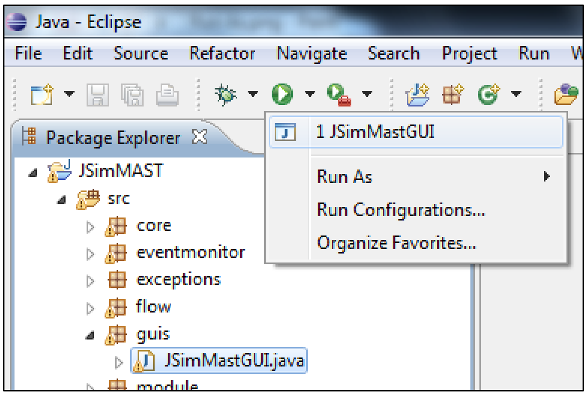


Figure 7. Invoking the predefined launch configuration