

POP

A TOOLSET FOR SIGNAL

POP Installation Guide (NOT UP TO DATE)

V0.0

	Author(s)	Checked by	Approval
Name	Members of the Espresso Team	Loïc Besnard ** Thierry Gautier Paul Le Guernic	Jean-Pierre Talpin
Company	INRIA	INRIA, **CNRS	INRIA
Department	Espresso Team	Espresso Team	Espresso Team
Date			
Visa			
Summary	Installation Guide of the SSME Platform, a front-end to the SIGNAL TOOLBOX under Eclipse. It must be up to date according the Eclipse policy. Do not distribute.		

Attention: la responsabilité des entreprises et des organismes ayant participé à l'élaboration de ce document ne peut en aucun cas être engagée en cas de dommages ou de pertes résultant de l'utilisation ou de l'exploitation des informations qui y sont contenues.

Disclaimer: Contractors participating to this report shall incur no liability whatsoever for any damage or loss which may result from the use or exploitation of information and/or Rights contained in this report.

Jan 30, 2015

Table of Contents

1 From a RCP bundle (Binary distribution).....	3
2 How to install SSME from the update site (Binary distribution).....	3
3 How to install SSME from the sources on Polarsys infrastructure(SOON).....	3
3.1 Getting the Eclipse environment.....	4
3.2 Installing EMF (Eclipse Modeling Framework).....	5
3.3 Getting the sources from the Polarsys infrastructure.....	5
3.3.1 About Git.....	6
3.3.1.1 Installing Git.....	6
3.3.1.2 Importing plugins from a Git repository.....	6
3.3.2 Import of the SSME plugins.....	6
3.3.3 Import of the SSME-SignalToolBox plugins.....	6
3.3.4 Getting the SignalToolBox sources.....	6
3.4 How to compile the Signal compiler under Eclipse.....	7
3.4.1 How to create a configuration for compiling SSME.....	7
3.4.2 How to compile the SSME environment.....	7
3.4.3 Testing your SSME environment.....	7
3.5 How to regenerate the plugins from the model.....	8
3.6 How to create an update site for SSME.....	9
3.6.1 Testing the generated site.....	9

This guide describes the SSME platform installation from the Polychrony web site on the Polarsys infrastructure (<http://polarsys.org/downloads/polychrony>). It describes how to install the platform from the RCP bundles (section 1), the update site (section 2) and from the sources(section 3).

1 From a RCP bundle

The RCP contains Eclipse + the SSME plugins.

Let suppose that you have already installed a Java Run-time Environment.

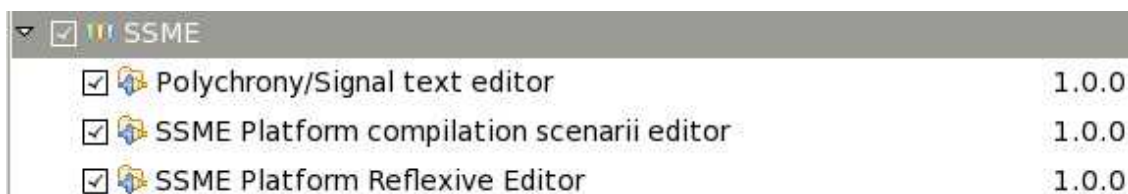
- *Download* the archive corresponding to your operating system from the Polarsys URL (SSME paragraph).
- Just *unzip* into your own location. It is ready to use.

2 How to install SSME from the update site¹

We suppose that you have a fresh Eclipse² installation (*Eclipse Classic*).

Under Eclipse

- Click on **Help ->Install New Software**.
- Click on **Add...** to add one of the following site:
 - <http://polarsys.org/downloads/polychrony/SSME/update>
- Select all the plugins



- Follow the installation.

You are now able to use the SSME environment.

3 How to install SSME from the sources on the Polarsys infrastructure.

TO BE COMPLETED -UNDER CONSTRUCTION.

To build the SSME environment, the first step consists in getting the Eclipse environment. To build the SSME environment from the sources, it requires:

- the installation of Eclipse Modeling Framework (EMF)
- the availability of the following sources
 - the plugins that define the SSME model,

¹The installation has been tested with Juno, Indigo, Helios Eclipse versions.

² Eclipse URL: <http://www.eclipse.org/downloads/>

- the availability of the plugins that define the connection between the SSME part and the functionality of the Signal toolbox,
- the availability of the “Signal toolbox” (the sources of the Signal Toolbox are not integrated under Eclipse but they must be accessible in your environment).

After, we describe

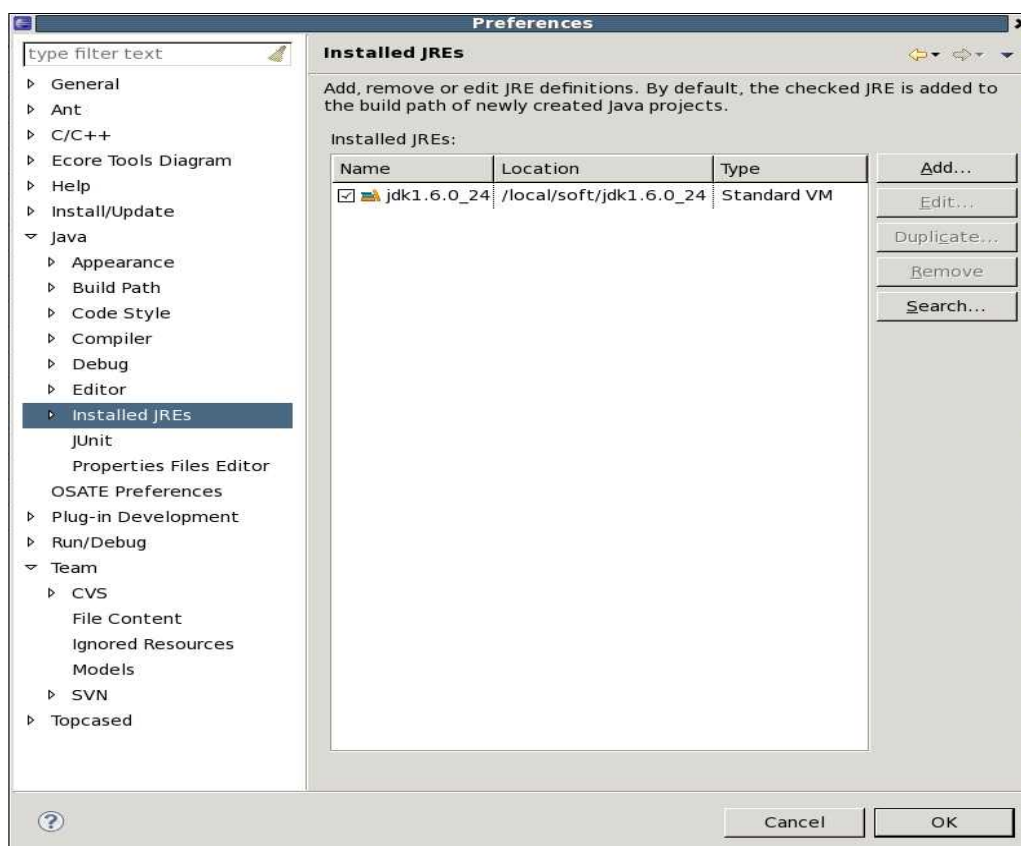
- How to compile the Signal compiler under Eclipse
- How to regenerate the plugins from the model
- How to create an update site for SSME

3.1 Getting the Eclipse environment

We suppose that you have a fresh Eclipse¹ installation (*Eclipse Classic* or *Eclipse Modeling Tools*).

Java JDK version under Eclipse

You have to be sure that you are using an Oracle JDK(1.5 or greater): in the **Window** menu, click on **Preferences**. Open the **Java** menu and click on **Installed JREs**:



If you are not using an Oracle JDK (ex: a JRE, or another JDK), you have to download it:

¹ Eclipse URL: <http://www.eclipse.org/downloads>

<http://www.oracle.com/technetwork/java/javase/downloads>

To add it to the **Installed JREs** list, click on **Add...** button and enter the path of the root of the JDK. Click on the box to select it and click on **OK**.

3.2 Installing EMF (Eclipse Modeling Framework)

If you have installed an Eclipse modeling Tools" version, this part is already installed (go to the next section).

The following site is already in the list of the available sites:

From <http://download.eclipse.org/releases/juno> (or *helios*, *indigo*) :



- Select the plugins
 - Modeling:
 - EMF - Eclipse Modeling Framework SDK
 - EMF Compare SDK
 - EMF Validation Framework SDK
- and follow the installation.

3.3 Getting the sources from the Polarsys infrastructure

UNDER CONSTRUCTION

The plugins that define the SSME model and the the plugins that define the connection between the SSME part and the functionalities of the Signal toolbox are available on the Git Polarsys repository((change **YOURNAME**)

git+ssh://YOURNAME@polarsys.org/gitroot/polychrony/polychrony.git

But, the source of the Signal ToolBox (under GPL Licence) are not available on the Polarsys infrastructure. It is available on the Polychrony Espresso site.

3.3.1 About Git

3.3.1.1 Installing Git.

If for some reason that perspective does not exist, you can install Git as follows.

- In the Eclipse tool, open menu **Help > Install New Software...**. A dialog window appears;
- For field *Work with:*, select **Indigo - <http://download.eclipse.org/releases/indigo>** and wait a while;
- Once the list of available components is filled, select the **Eclipse EGit** in the *Collaboration* category. Then click **Next >** twice;
- Accept the terms of the license agreement and click **Finish**;
- After the installation, Eclipse proposes to restart. Click on **Restart Now**.

Git uses a folder outside the workspace to store a local repository of the project. The location of this folder can be configured in the preferences.

- Open menu **Window > Preferences**;
- Select **Team > Git**;
- Change the value of *Default repository folder* if necessary, to your workspace;
- Click **OK**.

3.3.1.2 Importing plugins from a Git repository

- Select **File -> Import -> Git -> Projects from Git**
- Select **URL**
- Fill the field URL
- **TO BE COMPLETED.**

3.3.2 Import of the SSME plugins

Under Eclipse, import all the plug-ins of the **sub-directory SSME of the Polychrony git repository** in your workspace.

At this step, you must have errors on some plugins in your Eclipse editor.

3.3.3 Import of the SSME-SignalToolBox plugins

Under Eclipse, import all the plugins of the **sub-directory SSME-SignalToolbox of the Polychrony Git repository** in your workspace.

At this step, you must not have errors on any plugins in your Eclipse editor.

3.3.4 Getting the SignalToolBox sources

As said before, the source of the Signal ToolBox (under GPL Licence) is not available on the Polarsys infrastructure. It is available on the Polychrony Espresso site.

To get the source:

- *Download* the archive of the Signal ToolBox source available at

<http://www.irisa.fr/espresso/Polychrony/>

- *Unzip* this archive in your environment.

3.4 How to compile the Signal compiler under Eclipse

A **fragment** is a platform-specific plugin used in the SSME project to be able to work under different platforms. There are six fragments in the source code: **fr.irisa.espresso.ssme.polychrony.xxx.yyy.zzz**, where **xxx**, **yyy** and **zzz** are platform-specific identifiers. For example, the Linux x86_64 fragment is called: **fr.irisa.espresso.ssme.polychrony.gtk.linux.x86_64**.

In every fragment of the source code, there is an Ant script called *makePolychronySSME.xml*. You have to use the script located in the fragment corresponding to your platform. To compile this script, you have to modify (or create) an execution configuration.

3.4.1 How to create a configuration for compiling SSME

Under Eclipse:

- Right click on the Ant script *makePolychronySSME.xml*
- Click on **Properties**
- Go to the **Run/Debug settings** tab
- Click on **Edit** to modify a preexisting configuration, or **New** if there isn't any. In this last case, select **Ant Build** and click on **OK**.
- In the **ClassPath** tab, add *cpptasks.jar*¹ and *ant-contrib.jar*² with the **Add External JAR(s)** button.
- In the **makePolychronySSME** tab, you have to add this environment variable with the **New** button:
 - **SRC_POLYCHRONY** (absolute path to the root folder of the Signal toolbox).
 - **JDK_HOME** (with the complete path of the JDK)
- Close the environment.

3.4.2 How to compile the SSME environment

The file *OS_host.html* contains the definition of the name of the compiler and the linker for C/C++ languages and some options. You must update it for your environment.

To compile (Under Eclipse):

- right-click on the *makePolychronySSME.xml* script
- execute it with the **Run As --> Ant Build** option.

3.4.3 Testing your SSME environment

You can test your SSME environment by running it as an *Eclipse application*. A new

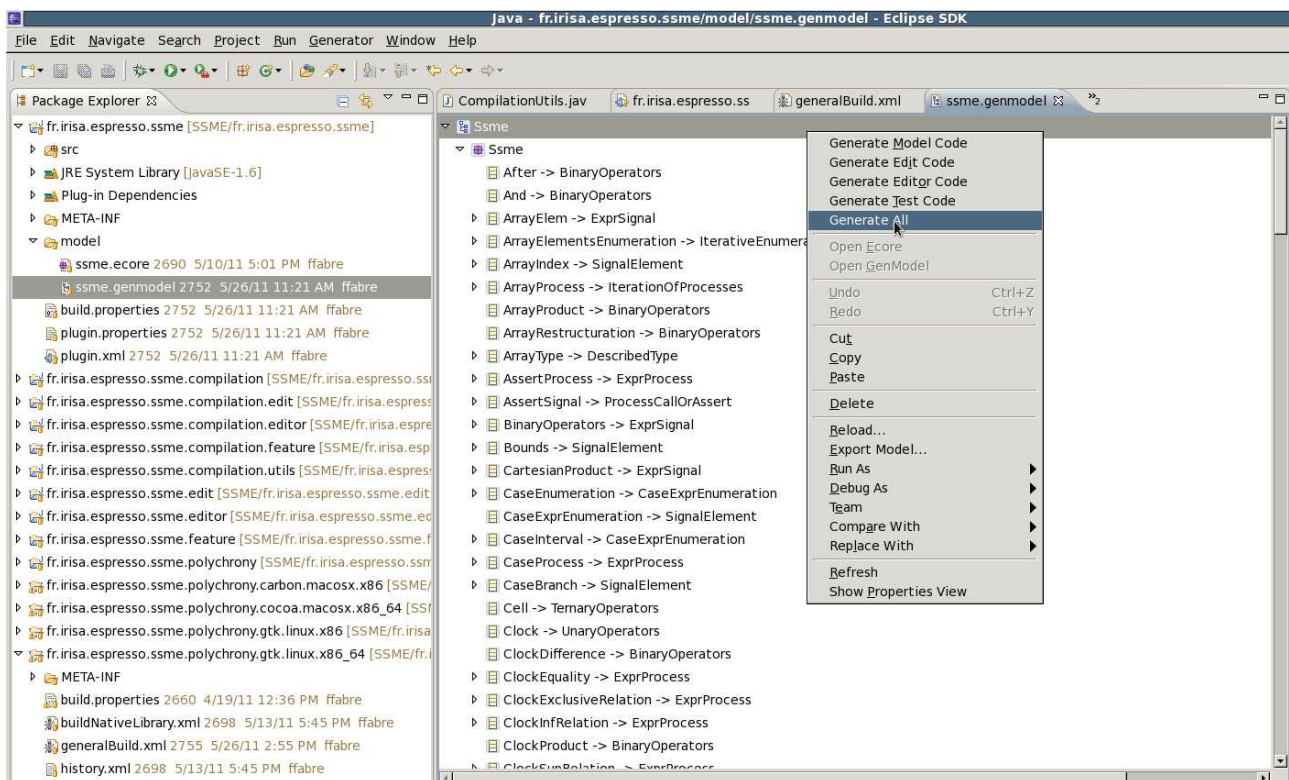
¹ *cpptasks.jar* is available on <http://mvnrepository.com/artifact/ant-contrib/cpptasks/1.0b5>

² *ant-contrib.jar* is available on <http://sourceforge.net/projects/ant-contrib/files/>

Eclipse is created in which the SSME plugins are installed.

3.5 How to regenerate the plugins from the model

It is possible to regenerate the plugins from the SSME meta-model. In the fr.irisa.espresso.ssme plugin, open the model folder to find the ecore model and the ssme.genmodel file. To generate the plugins from the .genmodel file, just click on the “generate all” option, as detailed in the following screenshot.



The following plugins will be regenerated:

- fr.irisa.espresso.ssme
- fr.irisa.espresso.ssme.edit
- fr.irisa.espresso.ssme.editor

If you don't want to overwrite a particular method in one of these plugins, you just have to prefix it with the “@generated NOT” Java annotation. For example:

```
/**
 * <!-- begin-user-doc -->
 * <!-- end-user-doc -->
 * @generated NOT
 */
public long getAST() {
```


3.6 How to create an update site for SSME

In the *fr.irisa.espresso.ssme.compilation.feature* feature, the option **Unpack the plug-in archive after the installation** must be activated for the *fr.irisa.espresso.ssme.compilation.utils* plugin. In the code of the *ExecuteScenarioJob.java* class, there are references to the *java.io.File* class that can't work if the plugin is packed into a .jar file.

In order to have the right version numbers for the fragments, you have to choose the **Copy versions from plug-in and fragment manifest** option with the **Versions** button of each feature.

Before building the site, you have to delete (if any) the *feature* and *plugins* directories and the *artifacts.jar* and *contents.jar* files of the *fr.irisa.espresso.ssme.site* plugin. Then, you have to use the **Build all** button in the *site.xml* file of this plugin.

The site has to be built on the three platforms: Linux, Windows and MacOS, for x86 and x86_64 versions. For example, if you're working usually on Linux x86, you have to build the site under Linux x86_64, Windows x86 & x86_64 and MacOS x86 & x86_64, and then copy the .jar OS specific archives (built on the *plugins* folder of the *fr.irisa.espresso.ssme.site* plugin) in the same *plugins* folder under Linux x86.

Finally, you have to delete the two *artifacts.jar* and *contents.jar* files, and to execute the Ant script called *GenerateMetadata.xml* to rebuild them, in order to take the fragments generated on the other platforms into account.

3.6.1 Testing the generated site

You can test your site by using a fresh Eclipse installation: you just have to follow the **How to install the SSME plugins (Section 2)** part of this tutorial. Instead of entering the Polychrony/SSME update site address, use the site you've just generated by clicking on **Local**.

Then you can test your installation with the examples provided on the Polychrony web site.