

POLYCHRONY
A TOOLSET FOR SIGNAL
(SME Platform)

Polychrony SME Installation Guide

V1.0

	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 SME Platform, a front-end to the SIGNAL TOOLBOX under Eclipse.		

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.

Jun 8, 2012

Table of Contents

1 From a RCP bundle (Binary distribution).....	3
2 How to install SME from the update site (Binary distribution).....	3
3 How to install SME from the sources.....	3
3.1 Getting the Eclipse environment.....	4
3.2 Installing Topcased.....	5
3.3 Getting the sources from the Polarsys infrastructure (all users).....	5
3.4 Getting the source on the Polychrony web site (all users).....	5
3.4.1 Import of the SME plugins.....	5
3.5 Getting the source from INRIA gforge (restricted to Espresso team).....	5
3.5.1.1 Installing SVN.....	6
3.5.1.2 Import of the SME plugins.....	7
3.5.1.3 Getting the SignalToolBox sources.....	7
3.6 How to recompile the Signal compiler under Eclipse.....	8
3.6.1 How to create a configuration for compiling SME.....	8
3.6.2 How to compile the SME environment.....	8
3.6.3 Testing your SME environment.....	8
3.7 How to regenerate the plugins from the model.....	9
3.8 How to create an update site for SME.....	11
3.8.1 Testing the generated site.....	11
4 Examples.....	11

1 From a RCP bundle (Binary distribution)

The RCP contains Eclipse + the SME plugins.

Currently, only win32 (32 bits) and Linux (64 bits) binaries distributions are available.

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

- Download the archive corresponding to your operating system from the link below
<http://www.irisa.fr/espresso/polychrony/SME>
- Just unzip into your own location. It is ready to use.

2 How to install SME from the update site¹ (Binary distribution)

We suppose that you have a fresh Eclipse (<http://www.eclipse.org/downloads/>) installation (*Eclipse Classic*). The installations of SME requires a part of Topcased plugins.

2.1 Installing Topcased

Under Eclipse

- Click on **Help -> Install New Software**.
- Click on **Add...** to add the following site:
<http://topcased-mm.gforge.enseeiht.fr/release/update-site3.6>
- **Select** everything, **except** :
 - Source Features:
 - Topcased build dependencies
 - Topcased SDK
 - Topcased Experimental :
 - C To UML Synchronization (Incubation)
 - Topcased Experimental (Incubation)
 - Uml2java Feature (Incubation)
 - Uml2rtsj Feature (Incubation)
 - Topcased Toolkit :
 - Topcased Environment
 - Topcased Requirement
 - Topcased SAM Editor
 - Topcased SysML Editor

2.2 Installing the SME plugins

- Click on **Help ->Install New Software**.
- Click on **Add...** to add the following site:
<http://www.irisa.fr/espresso/polychrony/SME/update>

¹The installation has been tested with Helios(3.6)/Indigo(3.7) eclipse versions.

- **Select** all the plugins
- **Follow** the installation.

You are now able to use the SME environment.

3 How to install SME from the sources

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

- the installation of Eclipse + Topcased
- the availability of the following sources
 - of the plugins that define the SME platform (SME model, the plugins that define the connection between the SME part and the functionalities of the Signal toolbox)
 - of the “Signal toolbox”: the sources of the Signal Toolbox are not integrated under Eclipse but they must be accessible in your environment.

The sources are available on

- the Polychrony website (all users)
- the Polarsys infrastructure (all users) **under construction**
- INRIA gforge **restricted to the Espresso team (developers)**

After, we describe

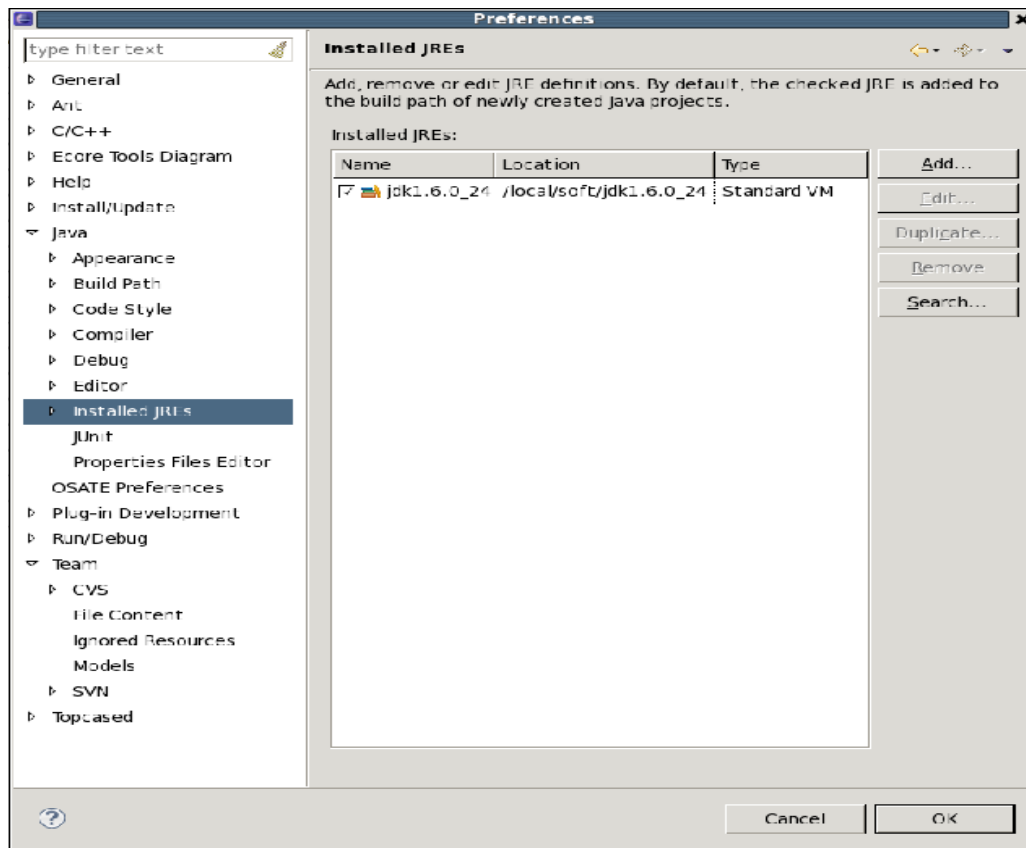
- How to recompile the Signal compiler under Eclipse
- How to regenerate the plugins from the model
- How to create an update site for SME

3.1 Getting the Eclipse environment

We suppose that you have a fresh Eclipse (<http://www.eclipse.org/downloads/>) installation (*Eclipse Classic, tested with Helios, Indigo*).

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: <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 Topcased

See 2.1

3.3 Getting the sources from the Polarsys infrastructure (all users)

UNDER_CONSTRUCTION

Then, goto the section 3.6

3.4 Getting the source on the Polychrony web site (all users)

To get the sources:

- **Download** the archives (select **SME** and **SignalToolBox** in the same line) from <http://www.irisa.fr/espresso/Polychrony/Sources>
- **Unzip** these archives in your environment.

3.4.1 Import of the SME plugins

Under Eclipse:

- **import** all the plug-ins of the **sub-directories of the SME** archive in your workspace.

Then, goto the section 3.6

3.5 Getting the source from INRIA gforge (restricted to Espresso team)

The sources of Polychrony are managed under the INRIA gforge (<https://gforge.inria.fr>).

To get the SME sources, you must have an account on INRIA gForge.
To be added to the polychrony project, send a mail to loic.besnard@irisa.fr

The Polychrony versions are managed by SVN.

3.5.1.1 Installing SVN

Under Eclipse, in the **Help** menu:

- Click on **Install New Software**.
- Click on the **Add...** button to add the following site: <http://openembedd.org/update>

Then select:

XTras:

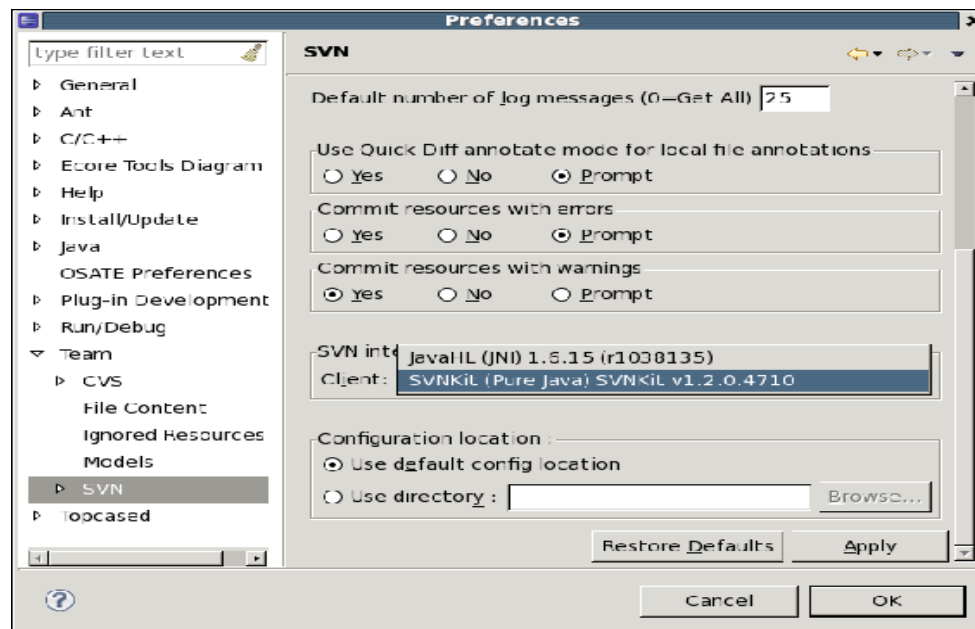
- Subclipse
- Subversion Client Adapter
- Subversion Native Library Adapter (JavaHL)
- SVNKit Client Adapter
- SVNKit Library

To use SVN with the INRIA gForge, you have to configure SVN. You have to use SVNKit instead of JavaHL, the default choice:

In the **Window** menu (see Illustration1)

- **click** on **Preferences**.
- **Open** the **Team** menu and **click** on **SVN**
- then **select** SVNKit instead of JavaHL

Illustration1: SVN configuration



Now we have everything to get the sources. The sources of the SME are composed of several parts:

- the plugins that define the SME model and the connection between the SME part and the functionalities of the Signal toolbox (for compiling SME programs).
- The Signal Toolbox

3.5.1.2 Import of the SME plugins

In the Eclipse package explorer, right click on **New --> Other**

- In the dialog box, select **SVN** then **Checkout projects from SVN** Click on **Create a new repository location**, then **Next**
- Enter this URL (change UserName)
svn+ssh://UserName@scm.gforge.inria.fr/svn/polychrony/SME_platform/trunk
- Your public/private SSH keys will be checked
- Import all the plugins of all the sub directories

3.5.1.3 Getting the SignalToolBox sources

To get the Signal Toolbox sources in your environment (outside Eclipse) in the sub-directory FOO, run the command

`svn checkout svn+ssh://UserName@scm.gforge.inria.fr/svn/polychrony/SignalToolbox/trunk FOO`

3.6 How to recompile the Signal compiler under Eclipse

A **fragment** is a platform-specific plugin used in the SME project to be able to work under different platforms. There are six fragments in the source code:

fr.irisa.espresso.SME.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.SME.polychrony.gtk.linux.x86_64**.

In every fragment of the source code, there is an Ant script called

makePolychronySME.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.

3.6.1 How to create a configuration for compiling SME

Under Eclipse:

- **Right click** on the Ant script *makePolychronySME.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 **Environment** tab, you have to add the following variable with the **New** button:
 - Name: **SRC_POLYCHRONY**
 - Value: the absolute path to the root folder of the Signal toolbox.
- **Close** the environment.

3.6.2 How to compile the SME 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 *makePolychronySME.xml* script
- **Execute** it with the "**Run As --> Ant Build**" option.

3.6.3 Testing your SME environment

You can test your SME environment by running it as an *Eclipse application*.

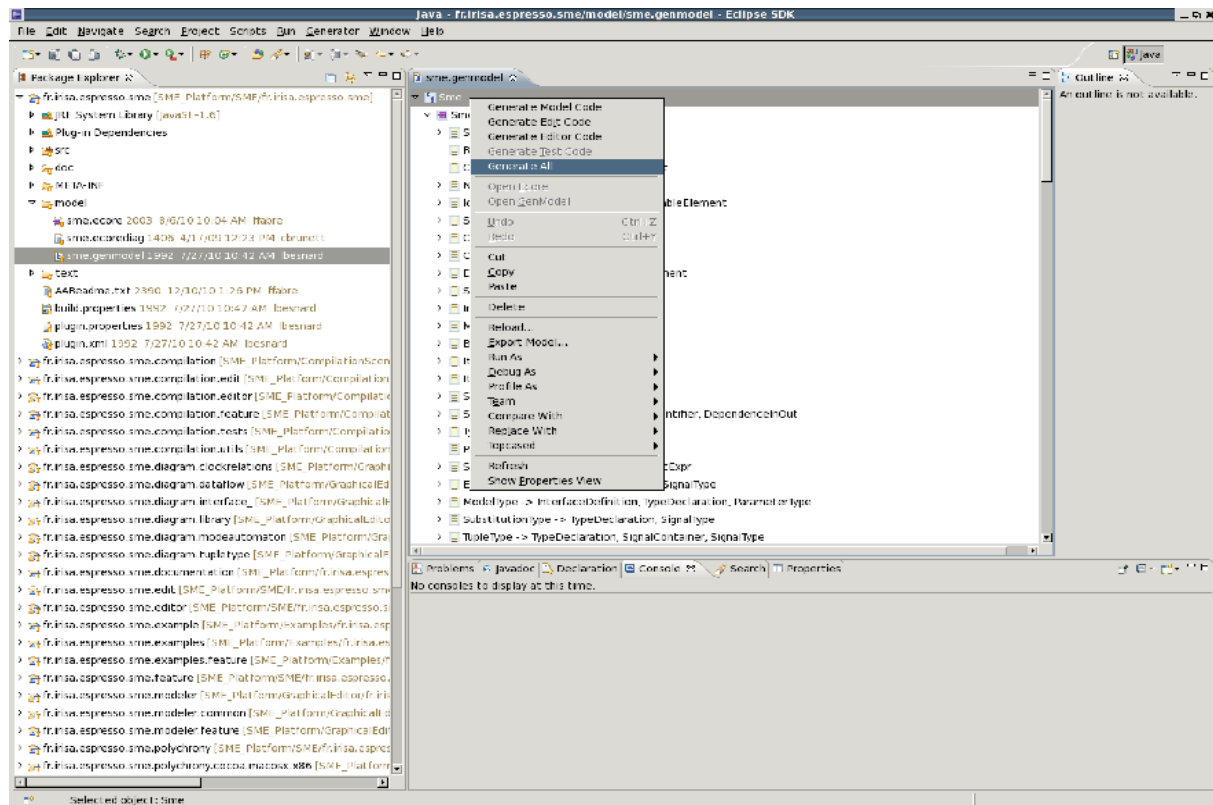
A new Eclipse is created in which the SME plugins are installed.

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/>

3.7 How to regenerate the plugins from the model

It is possible to regenerate the plugins from the SME meta-model. In the `fr.irisa.espresso.sme` plugin, open the model folder to find the `ecore` model and the `sme.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 :

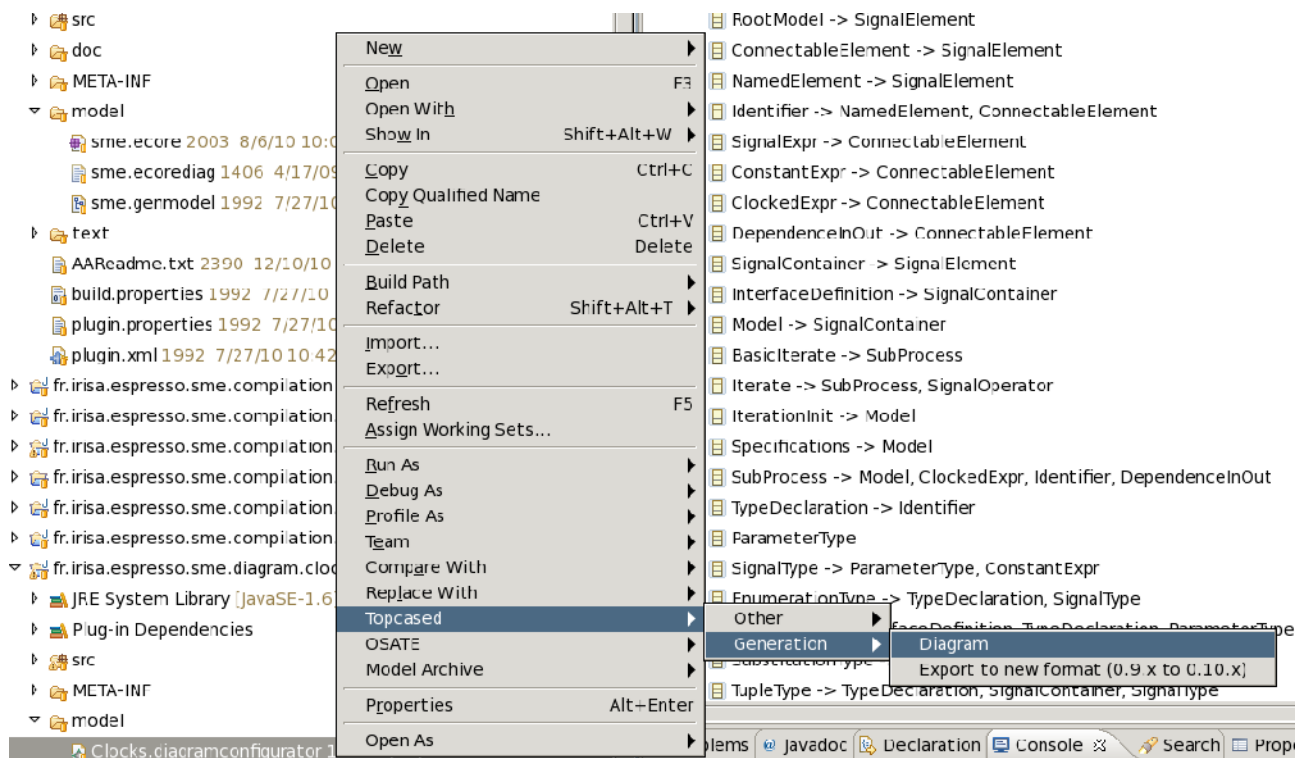
- `fr.irisa.espresso.sme`
- `fr.irisa.espresso.sme.edit`
- `fr.irisa.espresso.sme.editor`

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.

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

You can regenerate diagram editor plugins with the `*.editorconfigurator` file located

in each diagram editor plugin : right click on the *.editorconfigurator file, and click on Topcased --> Generation --> Diagram.



3.8 How to create an update site for SME

In the *fr.irisa.espresso.SME.compilation.feature* feature, the option

Unpack the plug-in archive after the installation

must be activated for the *fr.irisa.espresso.SME.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.SME.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.SME.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.

3.8.1 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 SME plugins (Section 2)*** part of this tutorial. Instead of entering the Polychrony/SME update site address, use the site you've just generated by clicking on **Local**.

4 Examples

To test your environment, you can

- **Download** the [Examples](#),
- **Import** them as **General->Existing Project into Workspace**,
- **Consult** the [SME user Guide](#).