# SME Installation Manual

V1.0.0 (July 11th, 2011)

# Table of Contents



# How to get SME sources

We suppose that you have a fresh Eclipse installation. Use Eclipse Helios ( v.3.6.2, edition *Eclipse Classic* : http://www.eclipse.org/downloads/packages/node/506 )

In the ***Help*** menu, click on ***Install New Software***. Install the following plugins :

From Openembedd - http://openembedd.org/update :

XTras :

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

From Topcased ( http://topcased-mm.gforge.enseeiht.fr/release/update-site3.6 ) :

Everything, except :

- Source Features :
  - Topcased build dependencies
- Topcased Experimental :
  - C To UML Synchronization (Incubation)
  - Topcased Experimental (Incubation)
  - Uml2java Feature (Incubation)
  - Uml2rtsj Feature (Incubation)

To use SVN with the INRIA gForge, you have to use *SVNKit* instead of *JavaHL*, the default choice. You have to change this in the following menu : ***Window*** --> ***Preferences*** --> ***Team*** --> ***SVN*** : here, select *SVNKit* instead of *JavaHL*.

To get the sources, you have to :

- Create an account on the forge (on the homepage).
- Contact polychronycontact@listes.irisa.fr to be added to Polychrony project.
- Configure Your SSH access (see the forge homepage).

Once your SSH keys are generated and accepted on the forge :

- In the Eclipse project 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
- Your public/private SSH keys will be checked
- Import all the plugins

## How to build the fragments

In every fragment of the source code, there is an Ant script. To compile this script, you have to modify (or create) an execution configuration.

In Eclipse :

- Right click on the Ant script
- 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 the ***ClassPath*** tab, add *cpptasks.jar* and *ant-contrib.jar* with the ***Add External JAR(s)*** button. In order to get *cpptasks.jar*, you can for example follow this link : http://mvnrepository.com/artifact/ant-contrib/cpptasks/1.0b5 In order to get *ant-contrib.jar*, you can for example follow this link : http://sourceforge.net/projects/ant-contrib/files/
- In the ***Environment*** tab, you have to add this environment variable :
  - ***SRC_POLYCHRONY*** (absolute path to the root folder of the Signal toolbox, containing the *PolychronyToolSet_setup* file.)

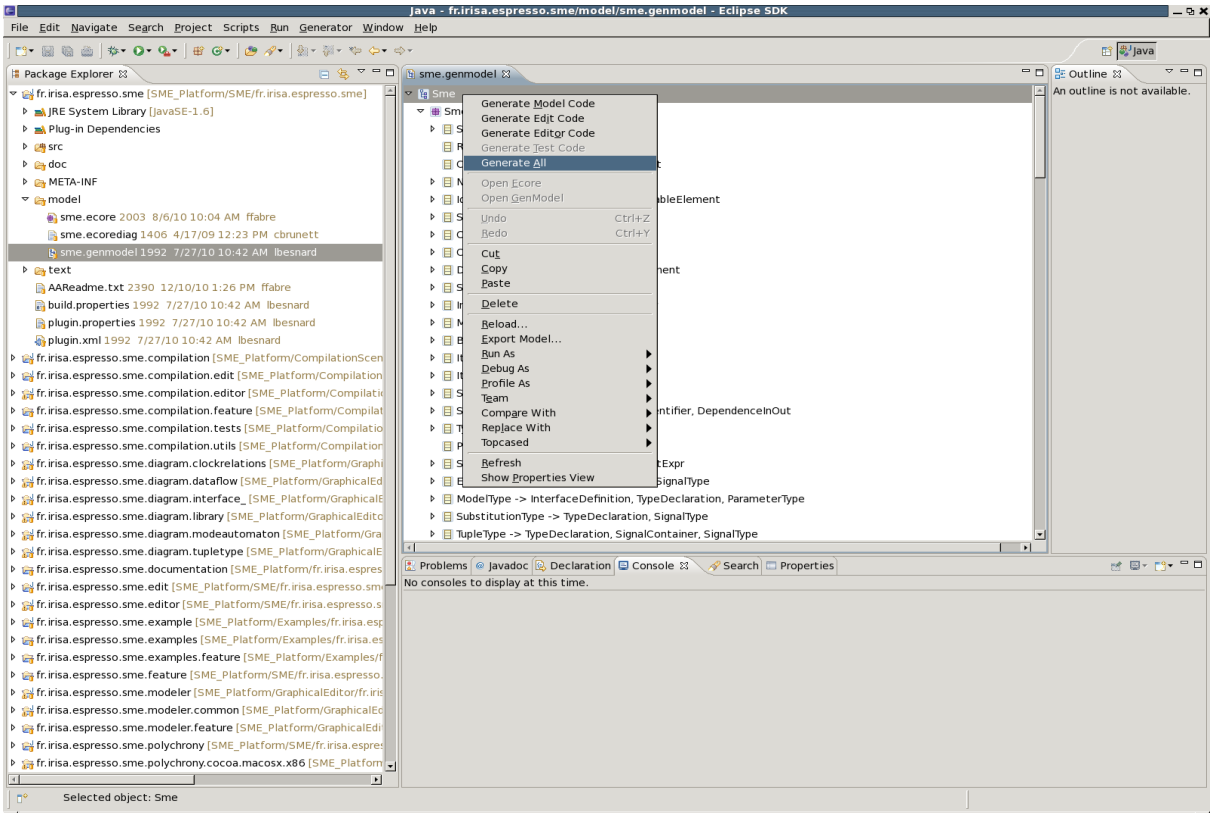
In the Ant script, the line *<property environment="evt" />* allows us to access the environment variables. The *${evt.SRC_POLYCHRONY}* command, for example, allows us to access the "SRC_POLYCHRONY' variable.

- Finally, execute the script with the ***Run As*** --> ***Ant Build*** option.

# How to regenerate the plugins from the model

It is possible te 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 :

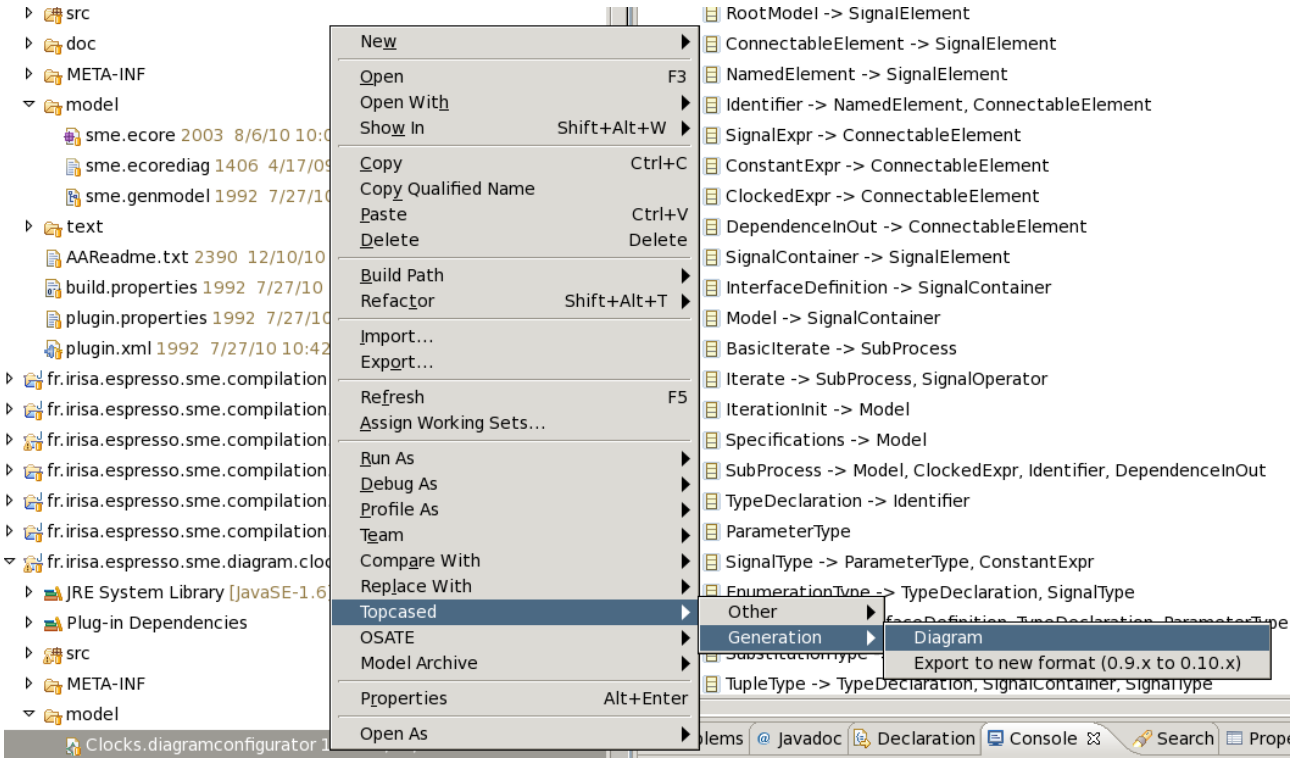
- fr.irisa.espresso.sme
- fr.irisa.espresso.sme.edit
- fr.irisa.espresso.sme.editor

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.



## 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.

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

# How to install the SME plugins

We suppose that you have a fresh Eclipse installation. Use Eclipse Helios ( v.3.6.2, edition *Eclipse Classic* : http://www.eclipse.org/downloads/packages/node/506 )

In the ***Help*** menu, click on ***Install New Software***.

From Topcased ( http://topcased-mm.gforge.enseeiht.fr/release/update-site3.6 ) :

- 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

From SME update site – www.irisa.fr/espresso/polychrony/update :

- Everything

You are now able to use the SME plugins.