# Polychrony: Signal ToolBox / Signal GUI install

Before installing the components, read the <u>License</u> terms and check the availability of <u>Required software</u>. The installation and use of the components suppose the agreement to the license. To get the source files and build your own toolset (SRC), you must proceed to the toolset <u>Extraction</u> and <u>Installation</u>, and then setup the toolset <u>Access</u>. You may also download an available Binary version (BIN) without the source code.

This notice primarily describes the commands for Unix system in a command window (the commands are almost the same in a Windows command window).

A detailed <u>Documentation</u> is available and can be partially rebuilt (DOC). An online user documentation is provided by the Signal GUI.

**POP platform**: The interface of the Signal ToolBox for POP (Polychrony on Polarsys Eclipse platform) is included in the Signal ToolBox component. To use the Signal ToolBox under POP, install it.

### 1) License (BIN+SRC)

Signal ToolBox and Signal GUI are free software under the <u>GNU General Public License</u>, <u>version</u> <u>2</u>. According to the license terms any user must have access to the source code, even if you (re)distribute the original or a modified binary version.

### 2) Required software (BIN+SRC+DOC)

(SRC+BIN) The following software are required and not provided to build a Polychrony environment:

```
• cmake - tcsh (all Unix based OS)
```

(SRC) The following software are required and not provided to build the executable files from the source files:

```
- C and C++ ANSI compilers (GCC or other)- JDK version 1.2.2 or more (for Signal GUI)
```

(DOC) The following external software are required and not provided to build the technical documentation:

```
- doxygen
- pdflatex, makeindex
```

(SRC+BIN) The following external software are required and not provided to use the corresponding outputs of the Signal ToolBox:

- Graphviz for displaying graphs described in XX.dot files
- Lustre for XX.lus files (coming soon)
- **C**, **C**++, **Java compilers** for XX.c, XX.cpp, XX.java output files
- Syndex for automatic code distribution of XX.sdx files

(SRC+BIN) The following companion software are required to use the corresponding outputs of the Polychrony toolset; they can be downloaded at this <u>Polychrony</u> site.

- SSME, the Eclipse frontend, for loading XX.ssme files
- Signal GUI, the Signal Graphical User Interface, for XX.gpk and XX.sig files
- Signal Toolbox for XX.sig files
- Sigali for XX.z3z files

## 3) Extraction (SRC+BIN)

• The Signal ToolBox is provided as a compressed archive file ( PolychronyToolset-SignalToolBox<Version><Suff>.tar.gz ) that contains a root directory called PolychronyToolset\_<Version> (where <Version> is the version of the Polychrony toolset). After an appropriate extraction¹ in XXX directory, you should get PolychronyToolset\_<Version> as a subdirectory of XXX.

Unix, MacOs
SRC+BIN gunzip PolychronyToolset-SignalToolBox<Version><Suff>.tar.gz
tar xvf PolychronyToolset-SignalToolBox<Version><Suff>.tar

• The Signal GUI (it must be uncompressed after the Signal ToolBox) is provided as a compressed archive file ( PolychronyToolset-SignalGUI
Version > Called SignalGUI
This archive must be uncompressed in the PolychronyToolset\_
Version > directory. If the Signal ToolBox is already installed, the Installation must be re-applied.

Unix, MacOS

cd PolychronyToolset\_<Version>
gunzip PolychronyToolset-SignalGUI<Version><Suff>.tar.gz
tar xvf PolychronyToolset-SignalGUI<Version><Suff>.tar

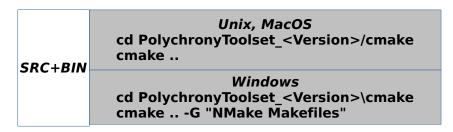
# 4) Installation (SRC+BIN)

**○** Make sure that the system variable "pK\_ROOT" is unset.

To install the extracted Signal ToolBox first configure your own distribution (SRC+BIN) and then generate the executable files (SRC):

- Configure (SRC+BIN): after unpacking the archive, move to the PolychronyToolset\_<Version>/cmake sub-directory, execute the command
  - **cmake ..** for Unix/MacOs,
  - **cmake .. -G "NMake Makefiles"** for Windows

<sup>1</sup>For windows, you can use a tool such that izarc (<a href="http://www.izarc.org/">http://www.izarc.org/</a>) to uncompress the archive.



• Generation (SRC): after configuring, move to the PolychronyToolset\_<Version> directory and execute the command "makePolychronyToolSet"

• To rebuild the documentation execute the following command line in the same directory:

### 5) Access (SRC+BIN)

Before using the Signal ToolBox first execute the following setup command in a fresh command window:

A short description is given by the command:

To use the Signal GUI execute the command **polychrony** and read the online user documentation.

# 6) Installation testing

Verify that the toolbox is correctly installed by testing your new Signal compiler: execute the command **signal** with appropriate parameters and options (**signal -h** to get the list of parameters and options). Some examples provided in the PolychronyToolset\_<Version>/Examples directory (each directory contains a AAREADME file) can be used for the test.

For the Signal GUI, run the **polychrony** command. Some examples provided in the PolychronyToolset\_<Version>/Examples directory) can be used for the test.

### 7) Documentation (SRC+BIN)

All the user documentation related to the Polychrony toolset including Signal ToolBox is available on the <u>distribution site</u>. This includes a general presentation of the Polychrony toolset architecture.

The source documentation (SRC only) is recursively accessible from:

Unix, MacOS
PolychronyToolset\_<Version>/doc/html/index.html
Windows
PolychronyToolset\_<Version>\doc\html\index.html

### 8) Uninstall

Simply delete the **PolychronyToolset\_<Version>** directory.

### 9) Error messages

When the pK\_ROOT is set, the installation may failed. The cmake command cannot produce a correct setup (**PolychronyToolset\_setup file**). To solve this problem, unset this shell variable and rerun the installing.

The cmake command fails when a required software (compiler) is not found.

### 10) Contacts

Loïc Besnard, email: Loic.Besnard@irisa.fr, CNRS, research engineer.

Member of the TEA team of INRIA Rennes Bretagne-Atlantique/IRISA.