

JavaBIP meets VerCors: Towards the Safety of Concurrent Software Systems in Java (Artefact Author README)

Simon Bliudze Petra van den Bos Marieke Huisman
Robert Rubbens Larisa Safina

Contents

AUTHOR readme of Artefact of: Verified JavaBIP: Deductive & Runtime Verification of JavaBIP Models	1
Summary	1
General usage	2
Files and folders relevant for compiling the tools	2

AUTHOR readme of Artefact of: Verified JavaBIP: Deductive & Runtime Verification of JavaBIP Models

This is the README intended for the authors of this artefact. It assumes knowledge of the general README (in the `README.md` file).

Summary

This artefact consists of two goals: primarily, to reproduce the results from the paper, secondarily, to aggregate and compile all sources into binaries needed for the primary goal. This readme is concerned with the second goal. As it is intended for the authors it is less polished.

Compiling VerCors & JavaBIP requires the Java & Scala ecosystem. This is difficult to package into a single zip, and hence we have not packaged it in this artefact. This also means the compilation part of this artefact is dependent upon having an internet connection, as well as all required repositories being available.

General usage

The design is that all scripts can be run in sequence, as indicated by their numbers, to reproduce all results from the paper.

The `00-*` and `1*-*` scripts are *preparation scripts*, meaning they prepare the artefact for submission. These are concerned with setting up the environment for compilation, cloning repositories, and compiling sources. They require build tools that must be installed via `apt`, as well as source repositories from `maven`, `apt`, and `sbt` to be available.

After running the `00-*` and `1*-*` scripts, the binaries (`.jar` and `.deb`) will be located in `binaries/`. These are self-contained binaries that will work on the FASE'23 VM. Before handing in the artefact, remove the `compiled/` folder, as it is no longer necessary. The packaging script (`13-*.sh`) also does this.

For information about the other scripts, see the main readme.

Files and folders relevant for compiling the tools

- `compiled/`: created by the scripts. Contains copies of the sources for use of compilation. When the compiling scripts are invoked, the `compiled/` folder will be used as scratch space.
- `AUTHORS_README.md`, `AUTHORS_README.pdf`: Readme containing instructions for compiling.
- `00-installCompileDependencies.sh`: installs dependencies needed for compilation and document generation using `apt`, such as `curl`, `git`, `maven`, `pandoc`, and `sbt`.
- `10-downloadSources.sh`: Clones all the repositories needed to compile JavaBIP, the Casino case study, and VerCors. This script also compiles the markdown files into PDFs, extracts the two case study examples from `sources/vercors-javabip/`, and creates a diff of the two case studies, so the changes are easily inspectable.
- `11-compileJavaBip.sh`: Compiles the Casino case studies into fat executable jars.
- `12-compileVerCors.sh`: Compiles VerCors into a debian package.
- `13-packageArtefact.py`: Packages the artefact into a zip, deleting folders that are not needed, and putting the zip hash into the artefact readme.
- `90-installAndRun.sh`: Shorthand script to install VerCors & run the case study with JavaBIP & VerCors. Note: Can only be run *after* the `91-setupAndCompile.sh` script. It has a lower number to ensure users will not accidentally call the `91-setupAndCompile.sh` script.
- `91-setupAndCompile.sh`: Shorthand script to install all compile dependencies, download all sources, compile JavaBIP & VerCors, and package the artefact into a zip.