
Compilation of Caltrop

A quick overview of the core source and compilation

I. Miller

document edition 1

ASTG Technical Memo
Programming Solutions Group
Xilinx
May 18, 2007

Contents

Contents	2
1 Overview	3
1.0.1 Core source tree overview	3
1.0.2 Dependencies	4
1.0.3 Compiling	4
1.0.4 Invoking the simulator	4

Chapter 1

Overview

This chapter details how to compile the *core* source tree of the caltrop repository. It is assumed that you are familiar with the Java programming language and general java compilation techniques.

1.0.1 Core source tree overview

The core section of the caltrop repository contains Java source code for the fundamental functionality related to caltrop. Additional ‘utilities’ are provided in the contrib section of the repository. Within the core source tree exists a full simulation infrastructure (with command line interface) including parser, expression evaluator/interpreter, platform, simulator engine and command line interface for invocation. In addition to the core sources, the repository provides required java libraries in the *lib* directory.

Core source packages

- util: Various utility classes providing basic and often reused functionality.
- cal: The CAL language parser.
- nl: The Network Language parser.
- hades: The interpretation and evaluation facilities used by the simulator.
- cli: The command line interface classes.

Provided libraries

- lib/source/java_cup: the JavaCUP parser generator. See <http://www2.cs.tum.edu/projects/cup/> for more information
- lib/source/ptolemy: Utilities from the Ptolemy project
- lib/saxon*.jar: The Saxon XSLT transformation engine. See <http://saxon.sourceforge.net> for more information

1.0.2 Dependencies

The packages detailed above maintain the following dependency relationships (excluding java.* dependencies):

- util:
- cal: util
- nl: cal, util, JavaCUP
- hades: cal, nl, util
- cli: hades, cal, nl, util, saxon

1.0.3 Compiling

The core source may be compiled with any Java 1.5 compliant compiler. A sample Makefile is included in `core/source/make`. This makefile will generate a jar file containing the generated classes from each of the above packages and the JavaCUP library. *NOTE: The saxon engine is not built into the jar file and must be included in your runtime classpath.* To build the jar file, simply 'cd' into the `core/source/make` directory and execute: `make jar`. The generated jar file will be located at: `core/caltrop.jar`

The core source may also be compiled within most common software IDEs. Your project compilation options should reflect the above packages and dependencies.

1.0.4 Invoking the simulator

Invocation of the simulator is accomplished by running java on the Simulator class. The classpath given to java must include both the `caltrop.jar` (generated as in the Compiling step above) and the Saxon jar files from the `core/lib` directory.

Executing the following command will display a help screen for the Simulator.

```
java -classpath caltrop.jar:lib/saxon8.jar:lib/saxon8-dom.jar  
net.sf.caltrop.cli.Simulator
```

Note that you must use the appropriate classpath separator for your platform (: in *nix and ; under windows) Further information can be found in the document: `GentleIntro.pdf`