

CVSim GUI Developer's Guide

Catherine Dunn

August 28, 2007

1 Acknowledgments

This implementation of CVSim is the work of Catherine Dunn with support by Thomas Heldt and contributions from Brandon Pierquet and Ali Saeed.

Previous implementations were written by Bob Sah (1983), George Moody (1985), Tim Davis (1989), Rama Mukkamala (2001), Thomas Heldt (2002), and by Eun Bo Shim and his students (2003). The circulatory model used in CVSim was created by Roger Mark, based on an analog model of JG Defares and colleagues. The CVSim model was elaborated by Thomas Heldt, Eun Bo Shim, Roger Kamm, and Roger Mark.

Development of CVSim was funded by the National Aeronautics and Space Administration (NASA) through the NASA Cooperative Agreement NCC 9-58 with the National Space Biomedical Research Institute.

2 Running CVSim via Java Web Start

The easiest way to run CVSim is using a web browser with Java Web Start installed.

2.1 Supported Platforms

Running CVSim via Java Web Start has been tested on the following platforms:

- Windows XP

- Mac OS X PPC
- Mac OS X i386
- Fedora Core 4, 5, and 7
- Red Hat Enterprise Linux 4

Running CVSim via Java Web Start is not supported on the following platforms:

- Solaris

2.2 Installing Java Web Start

In order to run CVSim from a web browser, you need to install Java Web Start. Java Web Start is part of the Java Runtime Engine (JRE).

If you are using Windows: Follow the instructions on the Sun website to install JRE 6.

If you are using Linux: Follow the instructions on the Sun website to install JRE 6. You also need to manually install the Java Plug-in in your web browser.

If you are using Mac OS X: Run Software Update from the System Preferences. Install Java for Mac OS X 10.4, Release 5.

The first time you run CVSim, you will see a security warning that asks you, “The application’s digital signature cannot be verified. Do you want to run the application”. Click “Run”. If you want to avoid that warning in the future, check the “Always trust content from this publisher” box. If you are using Internet Explorer, the security warning may be displayed behind your browser.

3 Building CVSim from the Source Code

3.1 System Requirements

In order to build CVSim, you need to have the following software installed:

- gcc, version 4.0.2 or higher (See <http://gcc.gnu.org>)
- Simplified Wrapper Interface Generator (SWIG), version 1.3.29 or higher (See <http://www.swig.org>)
- Java 2 Platform Standard Edition, version 5.0 Update 8 or later (See <http://java.sun.com>)
- GNU Make, version 3.80 or higher (See <http://www.gnu.org/software/make/>)

3.2 Obtaining the source code

The CVSim source code is available in a gzipped tar archive from <http://physionet.org/physiotools/>. Download this archive, and then unpack it by:

```
tar xfvz src.tar.gz
```

This will create a directory named `cvsim-NN` (where NN is the release number), containing subdirectories `devel` (code) and `doc` (documentation).

3.3 Compiling the Code

If you are using Linux: Navigate to the `devel` directory. Rename `Makefile.include.root.linux` as `Makefile.include.root`, and edit this file to reflect the location of your installation of gcc, SWIG, and Java. At the command line, type `make`.

If you are using Mac OS X: Navigate to the `devel` directory. Rename `Makefile.include.root.osx` as `Makefile.include.root`, and edit this file to reflect the location of your installation of gcc, SWIG, and Java. At the command line, type `make`.

If you are using Windows: Install and configure cygwin with all the required packages. Navigate to the `devel` directory. Rename `Makefile.include.root.windows` as `Makefile.include.root`, and edit this file to reflect your installation of gcc, SWIG, and Java. At the command line, type `make`.

4 Running the Code

To run the code, navigate to the `classes` directory and type:

```
java -Djava.library.path=../../lib edu.mit.lcp.CVSim
```

The `-Djava.library.path=../../lib` tells Java where to find the native C libraries. The `CVSim` class in the `edu.mit.lcp` package contains the `main()` method.

5 The Source Code

5.1 Directory Structure

What follows is a brief tour of the directory structure.

The top level of directories is

```
devel/      doc/
```

`devel/` is the main line of development. It's where all the code is. `doc/` is for project documentation, including this guide. In `devel/`, you will find a couple Makefiles, a shell script, and the following directories:

```
C/          java/      lib/       swig/
```

`C/` is for the C code (the model backend), `java/` is for the Java code (the GUI frontend), and `swig/` is for the SWIG interface that allows the two to work together. `lib/` is for the native C libraries generated from the C code and loaded by Java. The Makefiles are for building `CVSim` on Linux and Windows. The shell script, `osx_compile.sh`, is for building the application on Mac OS X. (Using a cross-platform build tool such as Ant might be better than using a combination of Makefiles and shell scripts. Alas, I did not have the time to get so fancy.)

Inside the `C` directory, you will find the source code for the 6-compartment model (`6_comp_model`) and the 21-compartment model (`21_comp_model`).

The `java/` dir is organized in the standard way with separate directories for source and class files (`src/` and `classes/` respectively) and a subdirectory structure mirroring the package name (`edu/mit/lcp`). In other words, the Java source files (`*.java`) are in `src/edu/mit/lcp/` and the class files (`*.class`) are in `classes/edu/mit/lcp`.