

DEVELOPING OOSIML SIMULATION MODELS

Using Codeblocks

Dr. José M. Garrido
Department of Computer Science

Updated November 2016

College of Computing and Software Engineering
Kennesaw State University

© 2015, J. M. Garrido

1 Introduction

There are several Integrated Development Environments (IDEs) that help developing programs written in C, C++, Fortran, and other programming languages. Eclipse is one of the most complete and powerful tools, it is mainly useful for Java programs and in general, it can be very slow. For C++, C, and Fortran programs, Codeblocks and CodeLite are faster, lighter, and more convenient tools to use.

This document briefly explains implementing OOSimL simulation models and C++ programs using CodeBlocks on Linux. For a more detailed introduction to CodeBlocks, refer to various websites with CodeBlocks tutorial and documentation. For a simple but complete tutorial, visit the web page:

`ksuweb.kennesaw.edu/~jgarrido/oopsim.`

2 Preparing CodeBlocks

The following sequence of steps involve the basic procedure for setting the appropriate options on Codeblocks for using the OOSimL translator then the remaining steps for editing C++ programs, compiling, and linking with the simulation library, *liboosimlc*.

2.1 Download the OOSimL Translator

The preliminary step required is the installation of the the OOSimL translator and the corresponding simulation library in the computer system running Linux. The 32-bit and the 64-bit versions of OOSimL translator are available on the web page mentioned previously.

1. Download the archive file `oosimlc.tar.gz` from the web page and extract all files from archive.
2. Select the executable file of the OOSimL translator (`oosimlc.out` for 32-bit Linux or `oosimlc64.out` for 64-bit Linux depending on the version of Linux your computer (32-bit or 64-bit). Rename `oosimlc64.out` to `oosimlc.out` in a 64-bit Linux. This executable file, the `compl` script file and the `oosiml.h` file must be stored on a folder such as `~/oosiml`.

2.2 Setting the Codeblocks Options

1. Start CodeBlocks and click on Create a new project.
2. On the top bar click on the Settings menu, then select Compiler. A dialog box appears. On the tab group, activate the Other Options tab, and click the button Advanced options located on the lower right of the dialog box.
3. Click the Yes button when the box *Edit advanced compiler setting?* appears.
4. On the new window that appears, click the '+' button and type `osl` in the dialog box. This indicates that the OOSimL files have an *osl* extension.
5. Type the Command line macro that will be used by CodeBlocks to execute the translator with a given OOSimL source file. For example: `~/oosiml/oosimlc $file`
Type the Generated files, which are C++ files, `$file.name.cpp` This assumes that the OOSimL translator (executable file `oosimlc.out`) is located in a folder named: `~/oosiml`
6. Click the Ok button located on the bottom of the window.

Codeblocks is now configured to recognize source files with an `osl` extension for editing and to execute the OOSimL translator.

2.3 Using Codeblocks

1. Select Console Application and click the Go button, which is located on the upper right corner of the window.
2. On the Console Application window, click the Next button. Select C++ language and click Next.
3. Type the project title (name), e.g. *Carwash*. In this example, the project will be created in folder: `/home/jgarrido/oosiml/models`. Click the Next button.
4. Select GNU GCC Compiler and click the Finish button.
5. Activate the left pane of the screen (Management), click the Projects tab. Remove the source file `main.c` by right-clicking on it and selecting delete file.

6. A new source file can be created by selecting File menu, then New, and Empty file. Click Yes to add this empty to the project. A new dialog window appears, type the name of the file with its `scl` extension. Now you can start editing this source file and when finished, save the file.
7. If one or more existing source files are to be included in the project, select the Project menu on the top bar, or right-click on the project name. Select Add files. Select the directory of the source to add to the project and select the OOSimL source file(s). Click Open.
8. The source files are now under Others and under the project name. Double-click on the desired source file to edit it further. The file now appears on the edit area of the screen.
9. Right-click on the current project name, or activate the Project menu in the top bar, and select Build options.
 - (a) On the tab Compiler settings, check Enable all compiler warnings.
 - (b) On the tab Linker settings, click the Add button to add a library to the project. Repeat to add all necessary object files and libraries. For a typical simulation model, at the minimum, the libraries `oosimlc` (in file `liboosimlc.a`), `pthread`, and `m` (standard math library) are necessary.
 - (c) On the tab Search directories and the tab 'Compiler', add the search directory for header files required by the source program while compiling. Assume the header files required by the programs are located in the folder `~/oosiml`.
 - (d) On the tab Search directories and the tab 'Linker', add the search directory of the libraries if needed. On Linux (Ubuntu), the external libraries are always stored in standard system directories. The local library `oosimlc` library can be stored in any location, for example `~/oosiml`. For this, click the Add button, then click (...) and navigate to the appropriate directory.
10. Build the project, which translates the OOSimL source file, then compiles and links the C++ files in the project. On the top bar, select the Build menu and select the Build option. The Build log appears in the lower pane of the Codeblocks screen.
11. To execute the program, select the Run option in the Build menu. A new screen appears with the results of the execution. After the program terminates execution, press the Enter key.

12. On the top bar, activate the File menu and select Close project. In the examples described previously, the project was created in the directory `~/oosim1/models`, Codeblocks creates several new subdirectories and the executable file is located in the directory `~/oosim1/models/Carwash/bin/Debug`.