Percolator in Eclipse (Ubuntu)

Mattia Tomasoni
Mattia.Tomasoni.8371@student.uu.se

August 27, 2010

Abstract

This tutorial will guide you through the necessary steps to get the Percolator source form the repositories, build it and import it into Eclipse. Debugging and version control (Git) will be possible from within Eclipse. I am running Ubuntu 10.04 LTS; Percolator is currently in its 1.14 version. Together with this tutorial:

- `build_ubuntu_eclipse_[XX]bit.sh`
- `install_gdb_printers.sh`

Get Eclipse

The latest to date Eclipse version for c++ developers can be downloaded at `http://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/heliosr`. Download the appropriate version, and run the executable `./eclipse`. No installation is needed.

Build Percolator as an Eclipse Project

We will now get Percolator from the repositories, ensure that the right libraries are installed and build it as a valid, debuggable Eclipse Project. This is done by invoking cmake with a special -G option that creates `.project` and `.cproject` files and and the "-D CMAKE_BUILD_TYPE" option set to "Debug".

- run the script `build_ubuntu_eclipse_[XX]bit.sh` from the location where you wish the percolator sources to be downloaded and built. You might be asked to insert the superuser password. The script comprises of four steps and will hopefully terminate with the following message: "SUCCESS! `buildDir` contains a valid Eclipse project.", where `buildDir` is some legal directory in your system.
- In Eclipse: File → Import → General → Existing Project into Workspace. In "Select root directory" put `buildDir`. Keep "Copy projects into workspace" unchecked and click "Finish".

The Project will be build automatically (no manual invocation of `make` and `make install`). You can monitor the build process from Eclipse's console. When this terminates the executables will be available in the directory "Binaries" of the "Project Explorer". To run (or debug) an executable, add the appropriate command line arguments to the run configuration.

For details on the creation of `.project` files with cmake, please refer to `http://www.vtk.org/Wiki/Eclipse_CDT4_Generator`.

Tweaking Eclipse (optional)

For better code writing, inspection and debugging I suggest the following:

- change the source hover background (set by default to black!). Windows → Preferences → C/C++ → Editor; scroll down the meny "Appearence color options" and select "Source hover background"; untick "System Default" and choose a more appropriate (light) color.
- Install STL support for GDB in order to get human readable prints (pretty-prints) of STL data structures during debugging: run the script `install_gdb_printers.sh` from any convenient location. For further details on STL support for GDB, please refer to `http://sourceware.org/gdb/wiki/STLSupport`
- change code auto-indentation in order to comply to Google C++ Style. Windows → Preferences → C/C++ → Code Style. Create a new profile with the "New" button and name it "google". A configuration window will open; under "General Settings", set "Tab policy" to "Spaces only" and "Indentation size" to "2". See `http://google-styleguide.googlecode.com/svn/trunk/cppguide.xml`

Version control in Eclipse

```
TODO

(http://wiki.eclipse.org/EGit/User_Guide)

Install EGit plugin (http://www.eclipse.org/egit/download/)
   help -> install new software ->
    work with: "http://download.eclipse.org/egit/updates"

Set up repository
   File -> Import -> Git -> clone
      URI: git+ssh://git@github.com/percolator/percolator.git
      (leave the rest of the forms as they are)
```