

## Wrangler Tutorial Exercise

### 1. Wrangler installation.

#### a. Linux and Mac OS X

- Make sure both Erlang (R11B-5 or later) and Emacs (Emacs22.1 or later) are installed.
- Download Wrangler from <http://www.cs.kent.ac.uk/projects/wrangler/>, or get it from the memory stick.

- In the Wrangler directory

```
./configure; make; (sudo) make install
```

- Add to your \$HOME/.emacs file

```
(add-to-list 'load-path "/usr/local/share/wrangler/elisp")
```

```
(require 'wrangler)
```

(if you use an install prefix other than 'usr/local', replace 'usr/local' with the correct dir.)

- Some of Wrangler's functionalities generate .dot files. If you would like to compile the .dot files and view the graph in Emacs, you need to add the following to .emacs file.

```
(load-file "/usr/local/share/wrangler/elisp/graphviz-dot-mod.el")
```

(Again, please replace 'usr/local' if necessary.)

- Wrangler requires a .erlang.cookie file in your \$HOME directory, please create one if it does not exist. The .erlang.cookie file should contain a single Erlang atom.

#### b. Windows

- Make sure both Erlang (R11B-5 or later) and Emacs (Emacs22.1 or later) are installed.
- Download installer from <http://www.cs.kent.ac.uk/projects/wrangler/>, or get it from the memory stick ...

- Run the installer.

- Some of Wrangler's functionalities generate .dot files. If you would like to compile the .dot files and view the graph in Emacs, you'll need to add the following to .emacs file.

```
(load-file "where-wrangler-is-installed/elisp/graphviz-dot-mod.el")
```

#### c. Eclipse + ErlIDE

- Make sure both Erlang (R11B-5 or later) is installed.
- On Windows systems, use a path without spaces in it.
- Install Eclipse 3.5, if you have not already.
- All the details as <http://erlide.sourceforge.net/>

## Wrangler Tutorial Exercise

### 2. Check Wrangler installation.

#### a. Emacs

- Start Emacs, open an Erlang file.
- M-x erlang-refactor-on or C-c, C-r to start Wrangler
- New menus: Refactor and Inspector should appear if Wrangler is installed properly.
- Customize Wrangler before starting to use:
- select Refactor -> Customize Wrangler, set the value of 'Wrangler Search Paths' as the directory where your Erlang code to refactor is.

#### b. Eclipse + Erlide.

- Start Eclipse and open an Erlang file, 'Refactor' should appear in the menu.

### 3. Basic refactorings.

Try some basic refactorings such as renaming, function extraction, generalisation, move function between modules, etc, with modules: pingpong.erl, s.erl, c.erl and server.erl.

### 4. Duplicated code detection and elimination.

- With modules pingpong.erl, smm\_SUITE.erl, try similar code detection and similar expression search. Note that the results returned are affected by the parameter values.
- Try to remove some of the clones found using refactorings: renaming and folding.

### 5. Module structure.

- generate module graph for the directory 'wrangler\_ex'.
- use Inspector -> Generate Module Graph to generate the module graph;
- use Graphviz -> Compile to compile the .dot file generated; and Graphviz->Preview to view the module graph in Emacs. A node colored in purple in the module graph means that the node is in a cycle.
- Use Inspector -> Cyclic Module Dependency to generate all the cyclic module dependencies.
- Check refactoring suggestions outputted in the \*erl-output\* buffer.
- Use Graphviz -> Compile and Preview to view the cyclic module dependency graph.
- Use Refactor -> Partition Exported Functions to partition the exports of module refac\_util.erl into groups.
- Move the three functions: write\_refactored\_files/5, write\_refactored\_files/6, and write\_refactored\_files\_for\_preview/3 to a new module, efac\_write\_file say. To do this, point the cursor to the export attribute which exports these functions, then select Refactor -> Move Function to Another Module.
- Move function get\_client\_and\_hrl\_file/2 to move wrangler\_modulegraph\_server. To do this, point the cursor to the function definition, then select the refactoring command.
- Generate the module graph again; there should be no more cycles.