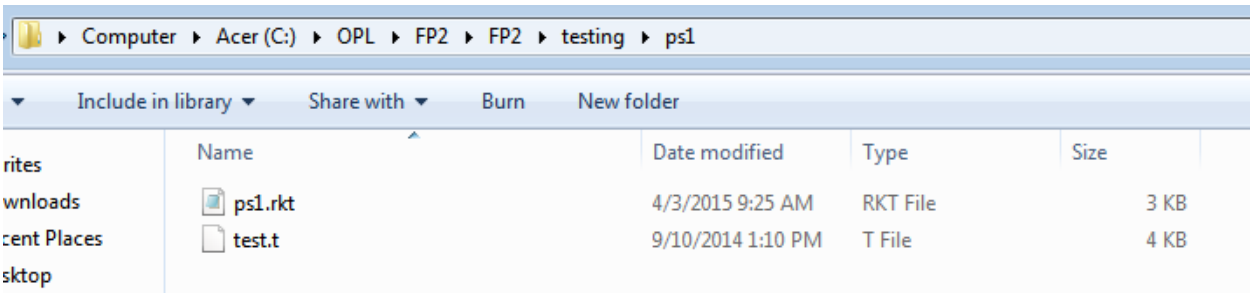
How Bottle-Racket is used

There are two important scripts that you'd actually use here:

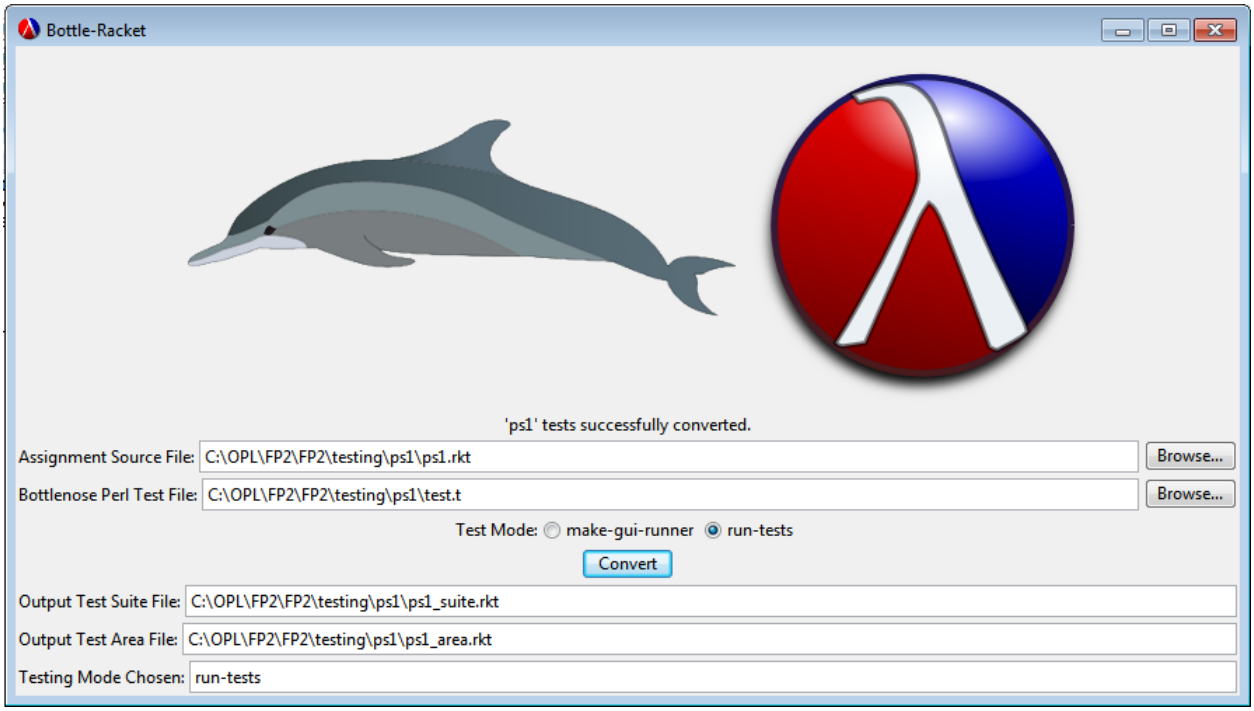
1.) `converter-gui.rkt`: The Bottle-Racket GUI which is used to create the test suite and area file for an assignment.

2.) `generate-results-runner.rkt`: Creates a script that runs the test area file generated from the script above, and saves the results into a nicely formatted email body as `test_email.txt`. This generated test running script is always going to be `test_script.rkt`.

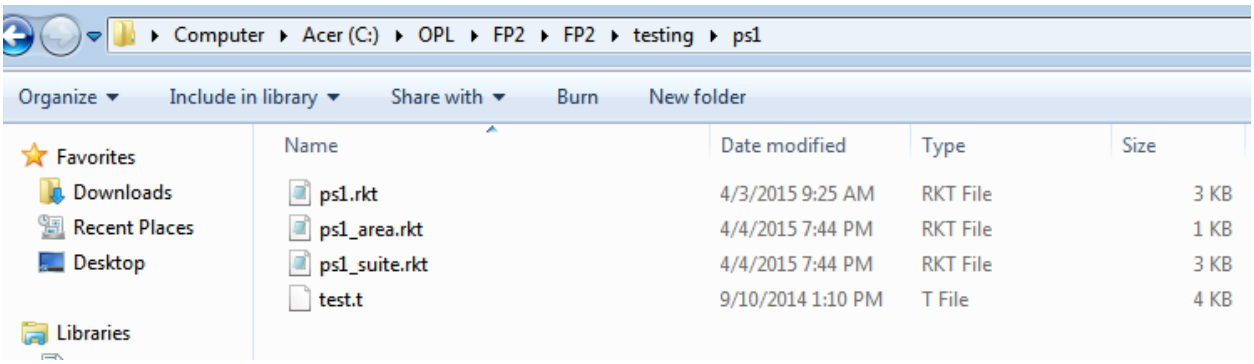
So first off, let's have a directory with the assignment source file and the Bottlenose Perl test file.



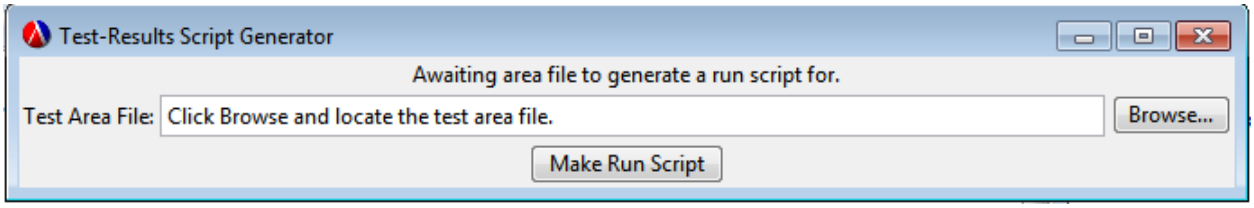
Then, run `converter-gui.rkt` to get the Bottle-Racket window. In my case, I was working with ps1. **Note that you MUST have a `(provide (all-defined-out))` statement at the end of the assignment source file or you will get unbound identifier errors.** Make sure the test mode is on `run-tests` as the textual interface is what allows us to capture test results. Then click the convert button and you should get updated text fields at the bottom of the window with the output test area and suite file, also indicating the test mode as run-tests.



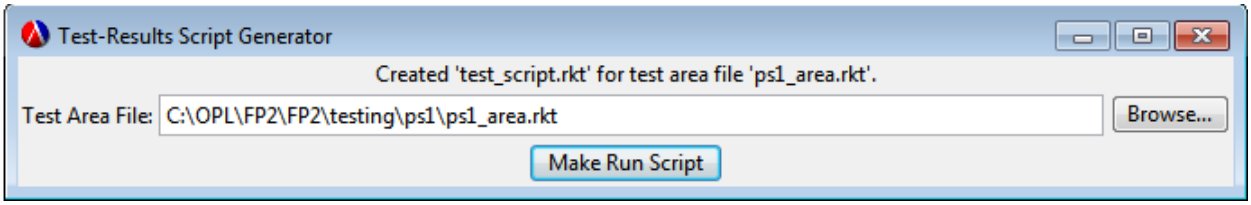
You should now see the generated area and suite files from Bottle-Racket in the same directory as your source files.



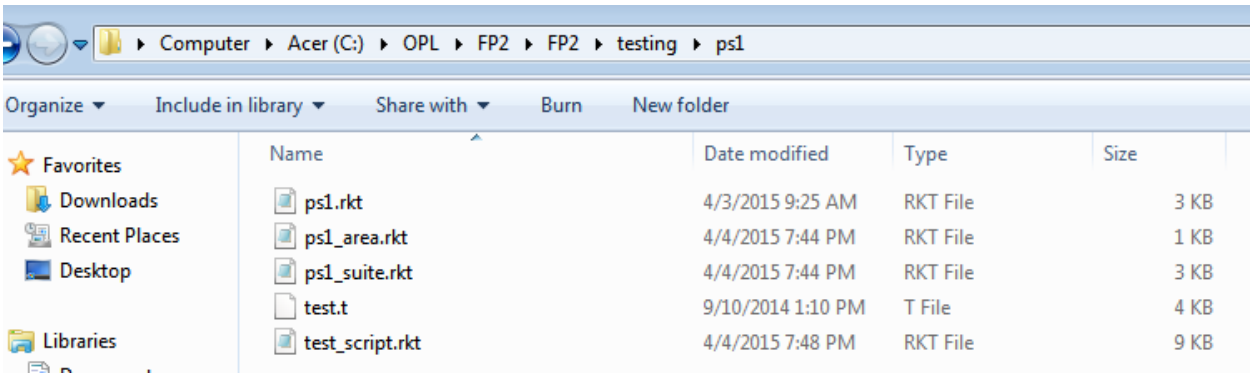
This is the point where you use `generate-results-runner.rkt` since you now have the test area file (in this case, `ps1_area.rkt`). Running `generate-results-runner.rkt` gives you a small simple window where you browse for that generated test area file to run.



After clicking the `Make Run Script` button, the message at the top of the window should update saying `test_script.rkt` was created.



You should notice `test_script.rkt` in the same directory as the area file as well.



Open `test_script.rkt` and run it. For reference, what happens in `generate-results-runner.rkt` is that the procedure definitions from `test-are-runner.rkt` are combined with a small "main" after which has a require statement for the area file which runs the tests. Then the redirected error output from any failed tests is saved into `test_results.txt`, this file is parsed for test case information, and then written out nicely into `test_email.txt` but right now it does not send the contents of that file through an

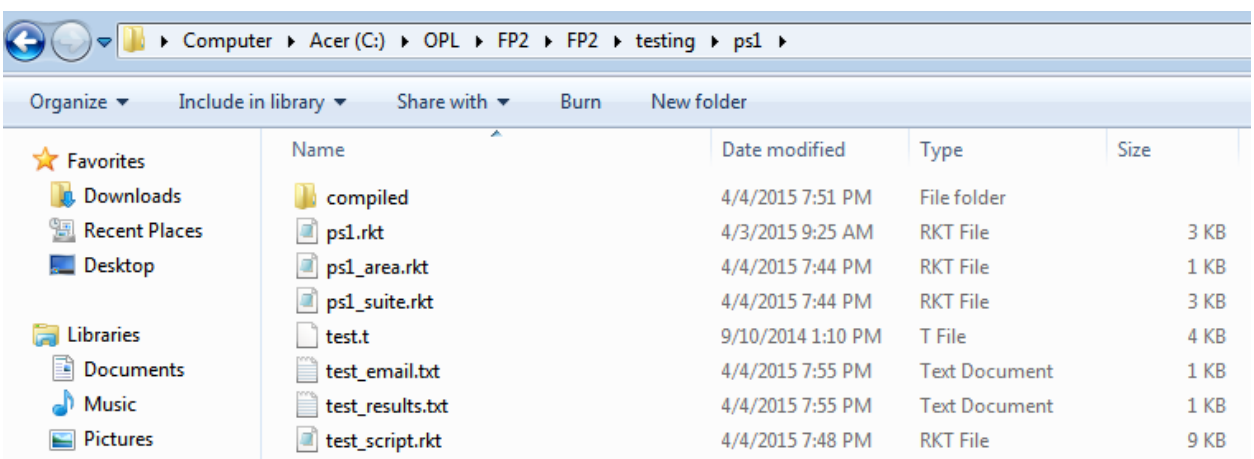
email. This is the important part of `test_script.rkt` which actually allows the `test_email.txt` file to be generated in the first place.

```
;; ********************************************************************
;; * MAIN: RUN THE SCRIPT
;; ********************************************************************

;; This line will run the tests
(require "ps1_area.rkt")

;; Read in the lines from the test results file
(define file-lines (file->lines "test_results.txt"))
(define suite-name (get-results-suite-name file-lines))
(define failed-case-lines-to-write (create-failed-cases-lines file-lines num-failed num-tests suite-name))
(display-lines-to-file failed-case-lines-to-write "test_email.txt" #:separator"\n")
```

After running `test_script.rkt` you should see two new files (ignoring the compiled folder). `test_results.txt` as mentioned previously is just the redirected error output from any failed test cases, and then `test_email.txt` is the file containing the email body to send out.



And then a picture of what `test_email.txt` looks like. In this case, 4 test cases failed and we can see some facts about how many cases passed and failed in this suite along with the specific tests that failed and why. In my case my `comb` function just returned 1 so this might not be a great example, but it should get the idea across.

C:\OPL\FP2\FP2\testing\ps1\test_email.txt - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

test_email.txt

```
>>> Results for test suite 'ps1'

-> Total: 30
-> Passed: 26 (87%)
-> Failed: 4 (13%)

> FAILED: '(comb 3 2)' in 'ps1_suite.rkt:21:26'
actual: 1
expected: 3

> FAILED: '(comb 4 2)' in 'ps1_suite.rkt:22:26'
actual: 1
expected: 6

> FAILED: '(comb 10 2)' in 'ps1_suite.rkt:23:27'
actual: 1
expected: 45

> FAILED: '(comb 93 37)' in 'ps1_suite.rkt:24:28'
actual: 1
expected: 118206769052646517220135262
```