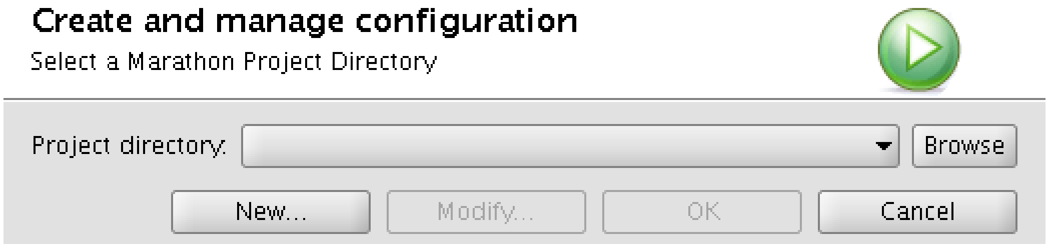
Using Marathon - GUI Acceptance Test Runner with isCOBOL™ Evolve

This document assumes that you have general knowledge on the use of Marathon, an open source project on SourceForge at http://sourceforge.net/projects/marathonman/

Here are the steps to create a Marathon project:

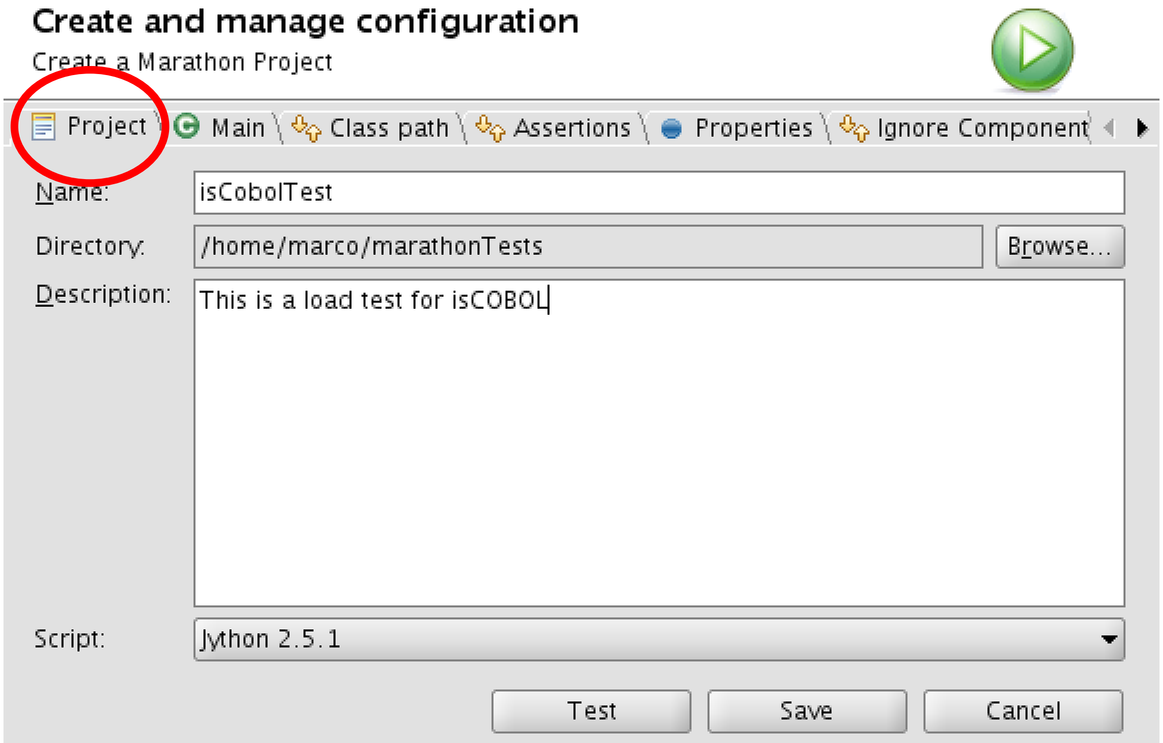
- install marathon according to its own documentation;
- create a directory in which all the configuration will be put; in my project I used `/home/marco/marathonTest`;
- run `marathon`: you'll get a window like the following one, choose 'New';



Create and manage configuration

Select a Marathon Project Directory

- you'll get a tabbed dialog: In this panel you enter the meta data about the project and also select the type of script you will be using with `marathon`;



Create and manage configuration

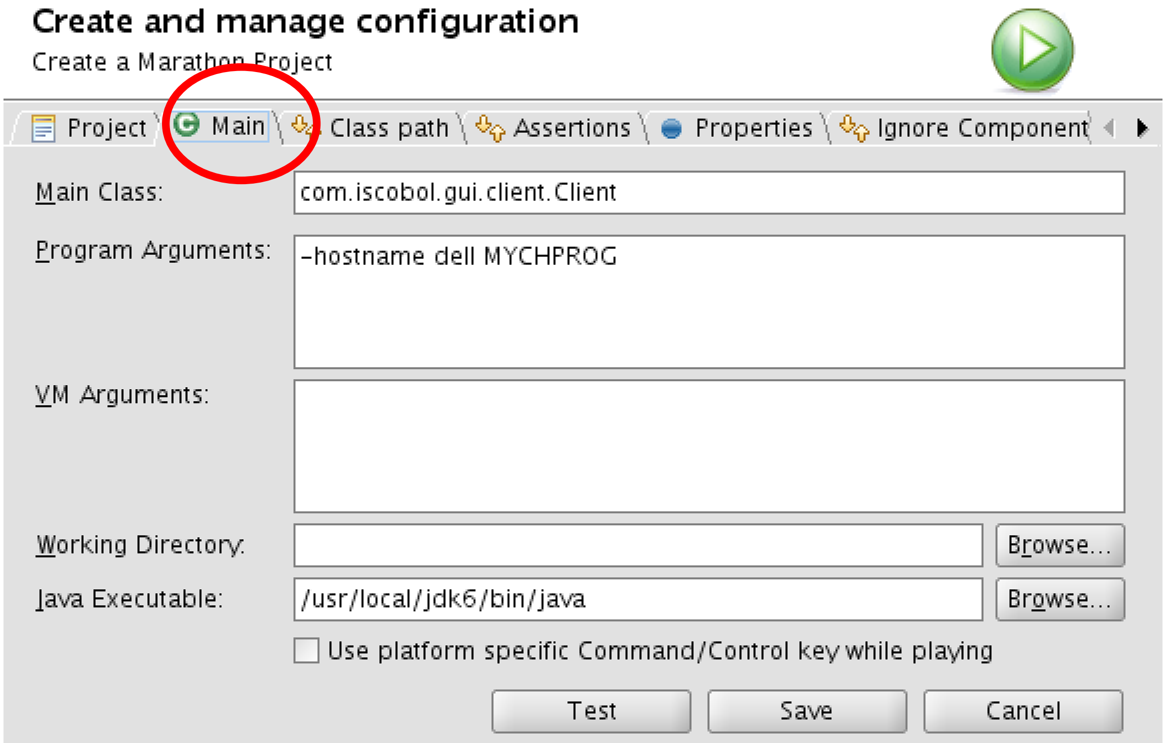
Create a Marathon Project

Name: isCobolTest

Directory: /home/marco/marathonTests

Description: This is a load test for isCOBOL

Script: Jython 2.5.1



Create and manage configuration

Create a Marathon Project

Main Class: com.iscobol.gui.client.Client

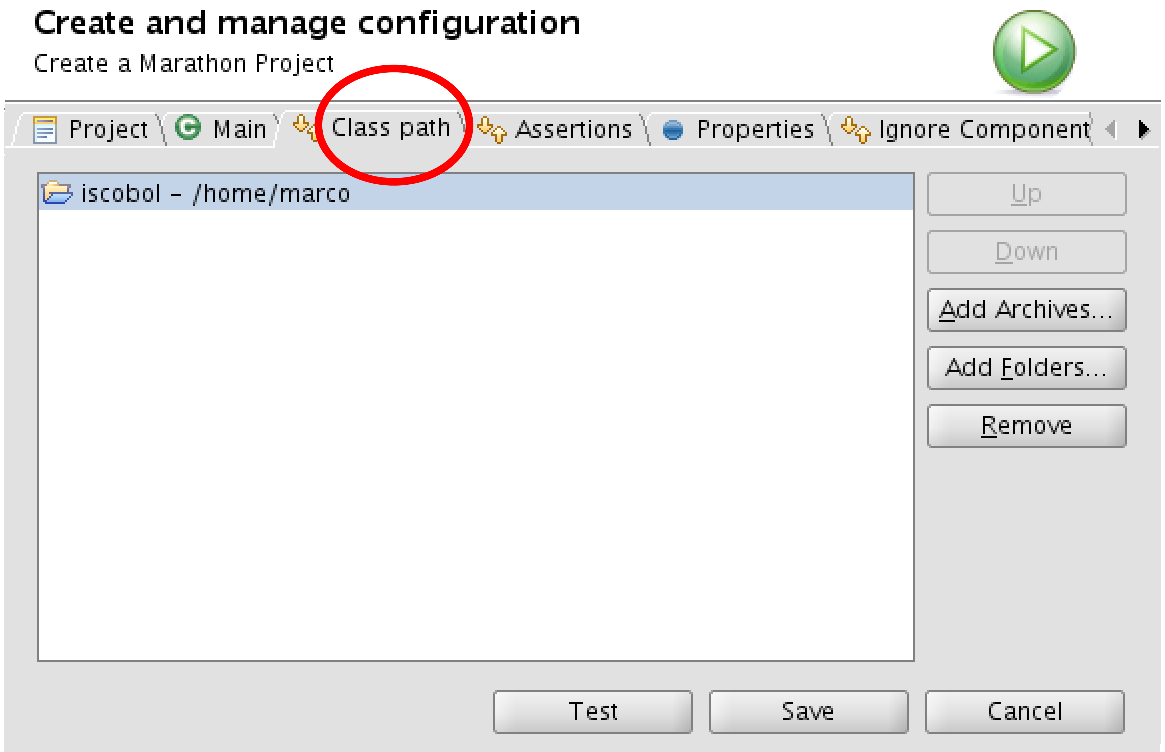
Program Arguments: -hostname dell MYCHPROG

VM Arguments:

Working Directory:

Java Executable: /usr/local/jdk6/bin/java

☐ Use platform specific Command/Control key while playing



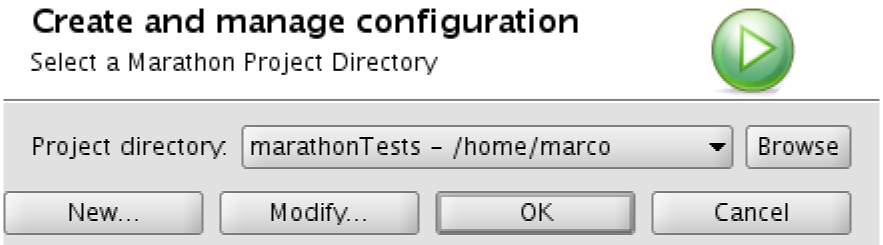
Create and manage configuration

Create a Marathon Project

iscobol – /home/marco

– After filling in the 3 tabs above, you can check to see if the data you specified is correct by pressing the 'Test' button, provided that the Application Server is started; if everything is OK, you'll see your application running;

– Press the 'Save' button, you will see the following dialog;

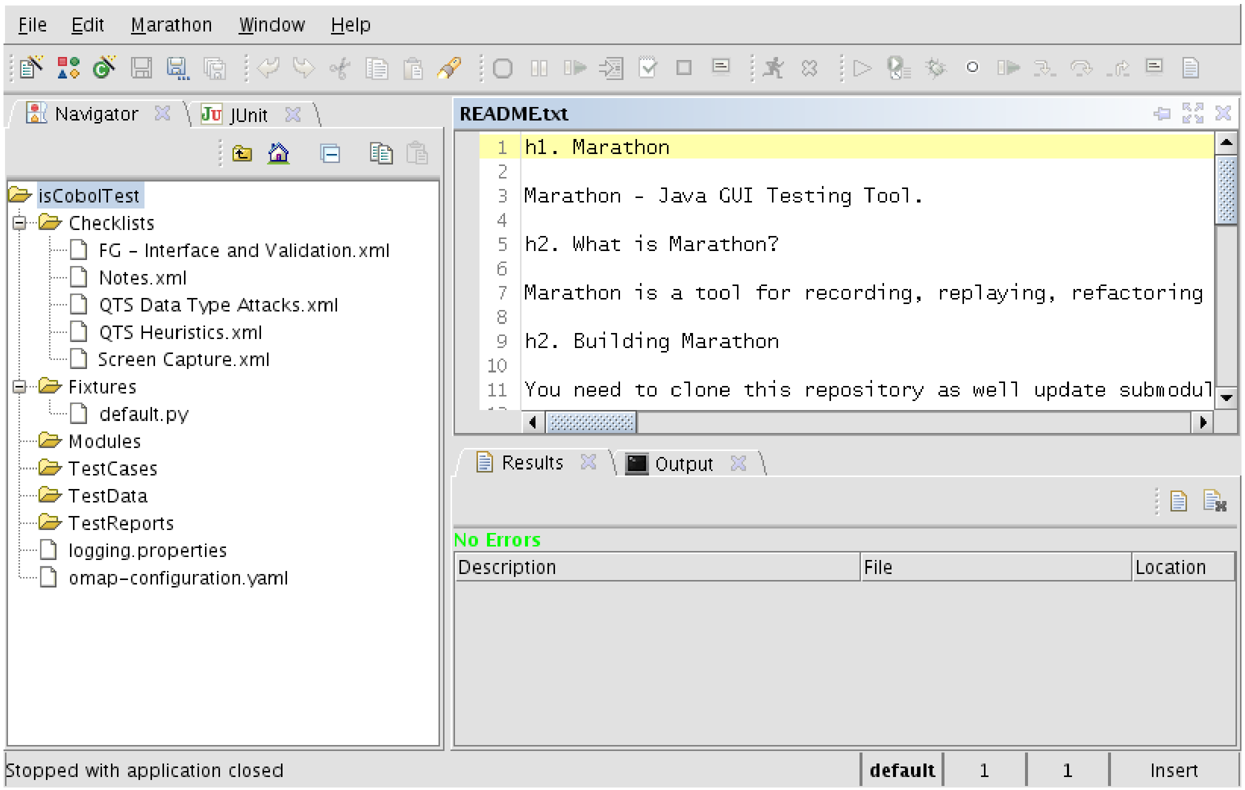


Create and manage configuration

Select a Marathon Project Directory

Project directory: marathonTests – /home/marco

– Press the OK button and proceed on the next window;

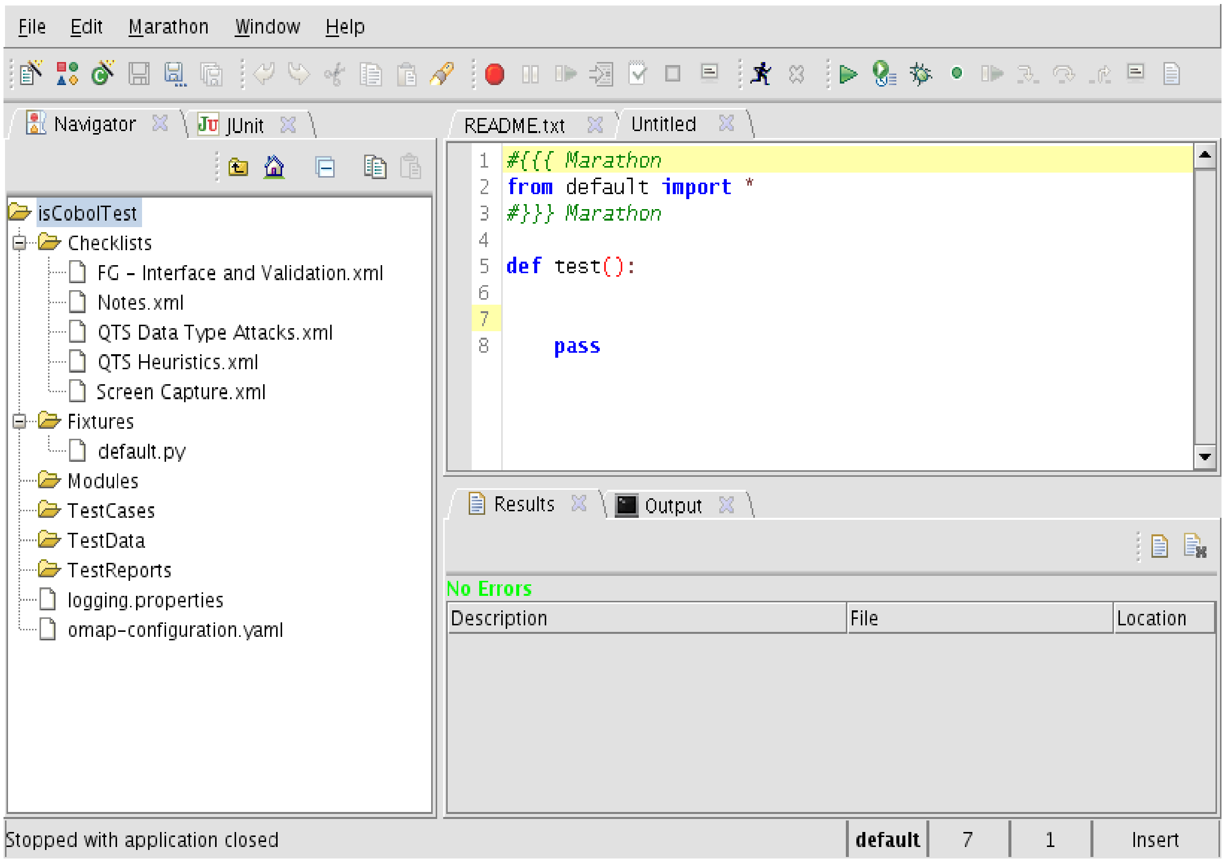


```
h1. Marathon

Marathon - Java GUI Testing Tool.

h2. What is Marathon?

Marathon is a tool for recording, replaying, refactoring

h2. Building Marathon

You need to clone this repository as well update submodul
```

No Errors

Stopped with application closed

- From the menu 'File' chose '*New->New Testcase*'

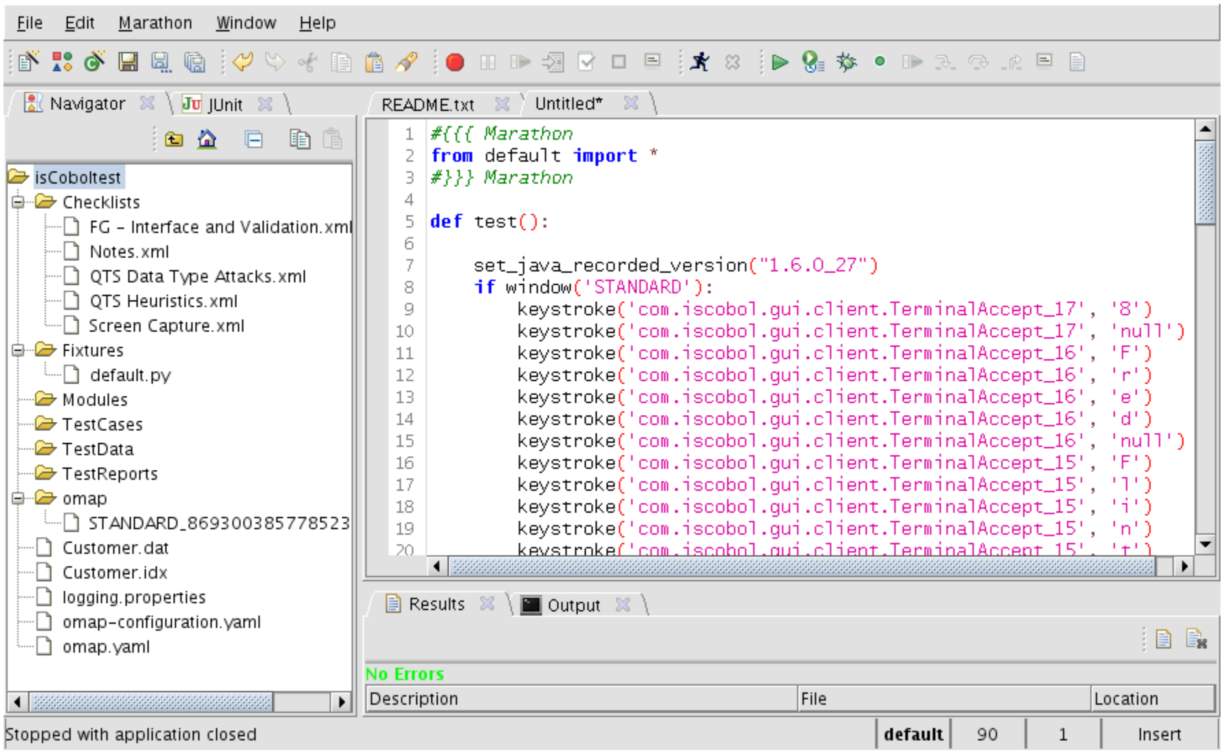


```
#{{{ Marathon
from default import *
#}}} Marathon

def test():


    pass
```

No Errors

Stopped with application closed

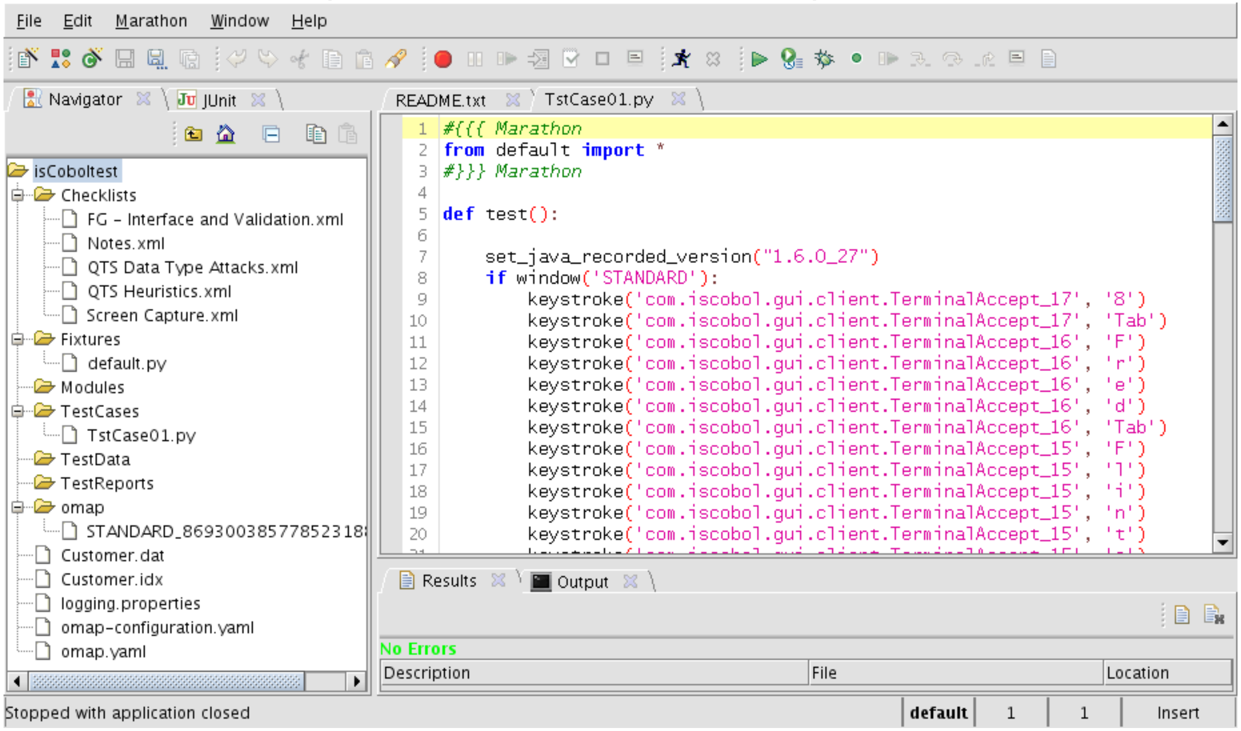
– At this point pressing the red button in the toolbar, your application will start and every action you do will be recorded; when you exit from the program you should get something like:



```
isCoboltest
  Checklists
    FG - Interface and Validation.xml
    Notes.xml
    QTS Data Type Attacks.xml
    QTS Heuristics.xml
    Screen Capture.xml
  Fixtures
    default.py
  Modules
  TestCases
  TestData
  TestReports
  omap
    STANDARD_869300385778523
  Customer.dat
  Customer.idx
  logging.properties
  omap-configuration.yaml
  omap.yaml
```

```
#{{{ Marathon
from default import *
#}}} Marathon

def test():

    set_java_recorded_version("1.6.0_27")
    if window('STANDARD'):
        keystroke('com.iscobol.gui.client.TerminalAccept_17', '8')
        keystroke('com.iscobol.gui.client.TerminalAccept_17', 'null')
        keystroke('com.iscobol.gui.client.TerminalAccept_16', 'F')
        keystroke('com.iscobol.gui.client.TerminalAccept_16', 'r')
        keystroke('com.iscobol.gui.client.TerminalAccept_16', 'e')
        keystroke('com.iscobol.gui.client.TerminalAccept_16', 'd')
        keystroke('com.iscobol.gui.client.TerminalAccept_16', 'null')
        keystroke('com.iscobol.gui.client.TerminalAccept_15', 'F')
        keystroke('com.iscobol.gui.client.TerminalAccept_15', 'l')
        keystroke('com.iscobol.gui.client.TerminalAccept_15', 'i')
        keystroke('com.iscobol.gui.client.TerminalAccept_15', 'n')
        keystroke('com.iscobol.gui.client.TerminalAccept_15', 't')
```

- The script on the right side is the real script to be used in the test; you can note some strings with the word '`null`': it seems a bug of the tool, those words must be changed to the word '`Tab`'.



```
isCoboltest
  Checklists
    FG - Interface and Validation.xml
    Notes.xml
    QTS Data Type Attacks.xml
    QTS Heuristics.xml
    Screen Capture.xml
  Fixtures
    default.py
  Modules
  TestCases
    TstCase01.py
  TestData
  TestReports
  omap
    STANDARD_86930038577852318
  Customer.dat
  Customer.idx
  logging.properties
  omap-configuration.yaml
  omap.yaml
```

```
#{{{ Marathon
from default import *
#}}} Marathon

def test():

    set_java_recorded_version("1.6.0_27")
    if window('STANDARD'):
        keystroke('com.iscobol.gui.client.TerminalAccept_17', '8')
        keystroke('com.iscobol.gui.client.TerminalAccept_17', 'Tab')
        keystroke('com.iscobol.gui.client.TerminalAccept_16', 'F')
        keystroke('com.iscobol.gui.client.TerminalAccept_16', 'r')
        keystroke('com.iscobol.gui.client.TerminalAccept_16', 'e')
        keystroke('com.iscobol.gui.client.TerminalAccept_16', 'd')
        keystroke('com.iscobol.gui.client.TerminalAccept_16', 'Tab')
        keystroke('com.iscobol.gui.client.TerminalAccept_15', 'F')
        keystroke('com.iscobol.gui.client.TerminalAccept_15', 'l')
        keystroke('com.iscobol.gui.client.TerminalAccept_15', 'i')
        keystroke('com.iscobol.gui.client.TerminalAccept_15', 'n')
        keystroke('com.iscobol.gui.client.TerminalAccept_15', 't')
```

- At this point, pressing the green arrow in the toolbar you should see your program running automatically.
- Save your script, you can edit it later, if you need it, with your favorite editor too.
- In order to execute the load test, you can run marathon many times from the command line specifying the -batch option and the directory containing your project, e.g.

```
marathon -batch /home/marco/marathonTest
```

I was able to run 10 marathon tests on my XP machine with 1 Gb of RAM.

Note that you can create a script by hand too or you can use a script created from a local application in order to feed a client server application.