# Introduction

This manual contains description how to install and launch the McGrawHill test automation framework.

# Tools And Project sourses

The project sources and other data is located at tfs <http://teg-ts.tegrity.local:8080>

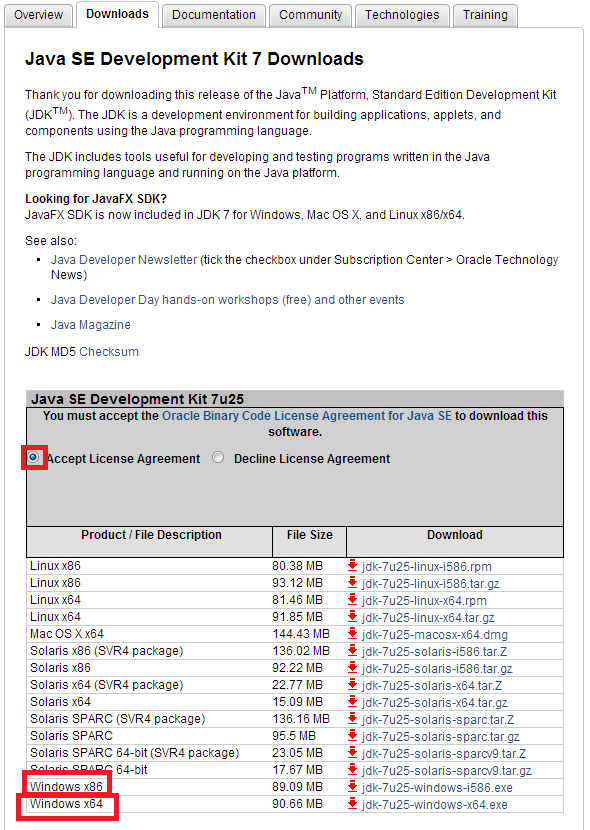
<project folder>: $/Tests/MHCampus/automation

<documents folder>: $/Tests/MHCampus/docs

To start work with the framework the following tools should be installed:

* JDK: you may find it here <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

To download click ‘Accept user terms’, and click the link to download the correct version for your system (See screen)



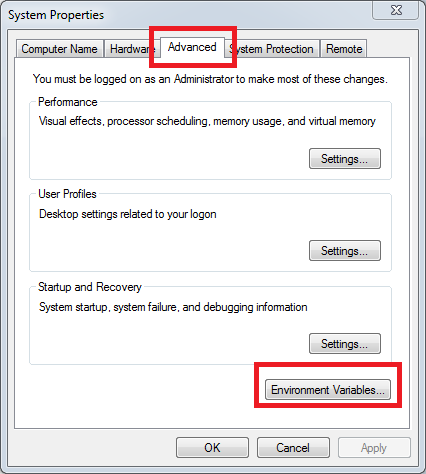
* Maven: you may find it here <http://www.eu.apache.org/dist/maven/maven-3/3.1.0/binaries/apache-maven-3.1.0-bin.zip>

Eclipse ide: use Eclipse ide you donloaded before

# Installation and Running from Command line

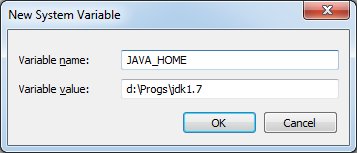
1. Java installation

* Install JDK.
* Go to windows environment variables: Press combination Win+Pause, on the opened window click change settings link, you’ll see System properties window (see screen)

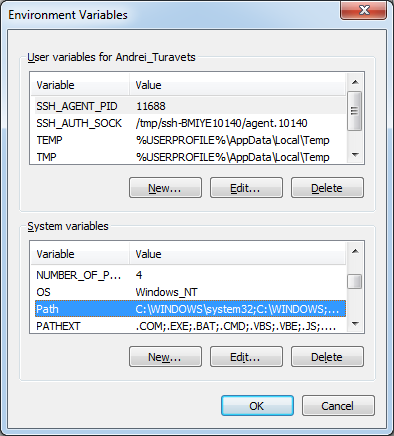


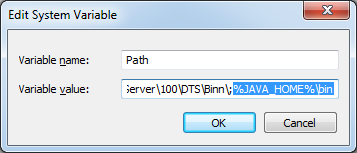
Go to ‘Advanced’ tab and click environment variables.

Under ‘System Variables’ click ‘New’ button. Fill variable name as ‘JAVA\_HOME’ and variable value as the path to your installed jdk (See screen below)

.

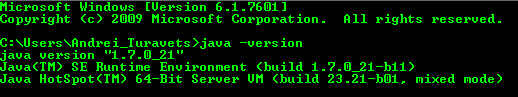
Click ‘OK’, then under ‘System Variables’ find ‘Path’ variable (See screen).



Click ‘Edit’ and to the end of ‘Varible value’ add <%JAVA\_HOME%\bin> to the end 

After completing check that java installed correctly: go to command line, and execute command java -version

You’ll see the response something like this



1. Maven installation.

Unzip Maven archive to some <maven folder> (‘e.g. d:\Progs\maven’).

Add System variable:

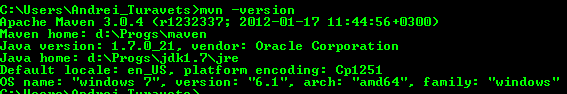
Variable name: M2\_HOME

Variable value: <maven folder>

Edit ‘Path’ variable: Add <%M2\_HOME%\bin>

After completing check that java installed correctly: go to command line, and execute command mvn -version

You will see the response something like this



1. Running tests from command line

Before running tests make sure, that <project folder>\mcgraw-tests\src\main\resources\ base.properties has correct data:

The server run pathes should be uncommented and local run pathes should be commented:

#Rampart path for server run

blackboard.api.rampart.path=..\\..\\..\\dependencies\\axis2-1.5.6\\repository\\

#Rampart path for local run

#blackboard.api.rampart.path=..\\dependencies\\axis2-1.5.6\\repository\\

#Path for server run

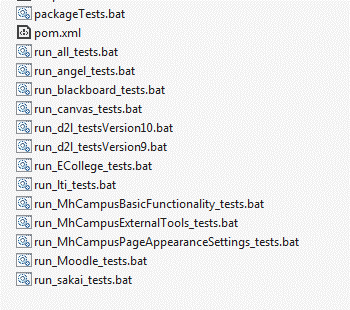
tegrity.deleteinstance.path=..\\..\\..\\dependencies\\DeleteInstance\\DeleteInstance.exe

#Path for local run

#tegrity.deleteinstance.path=..\\dependencies\\DeleteInstance\\DeleteInstance.exe

Go to <project folder>, double click at packageTests.bat will create executable jar file to launch our tests. After these actions, you may proceed to run the tests.

In the <project folder> you may find .bat files to run different test suites(See screen below)



To run all test double click on run\_all\_test.bat (Note: All tests run in about 8 hours)

To run exact suite just double-click on the suite you want to run.

1. Report

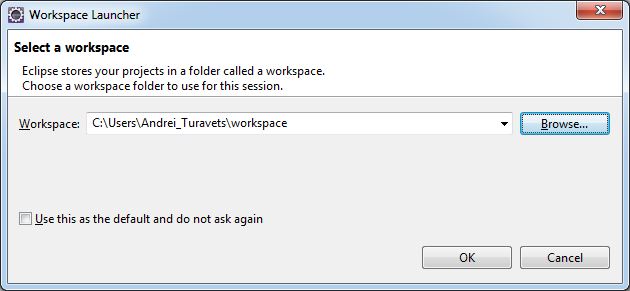
After the test run is finished, open file <project folder>\mcgraw-tests\target\mcgraw-tests-1.0-McGraw Selenium Tests\test-output\html\index.html

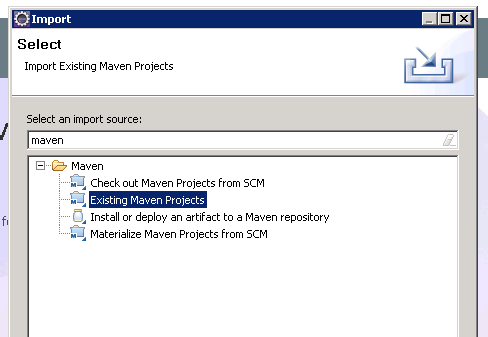
This is your report (Note: after test run the old report will be deleted and report for current test run is created)

# Eclipse installation and running

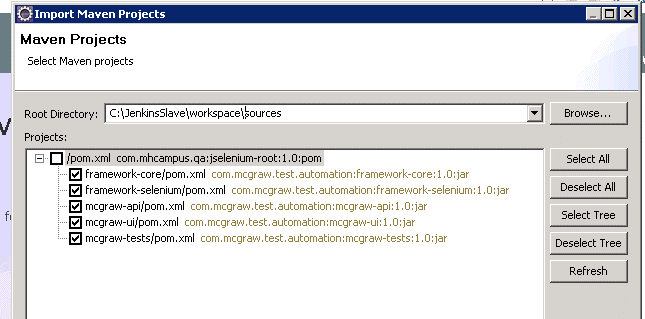
After eclipse is downloaded, unpack it to some <eclipse folder>.

In <eclipse folder> launch eclipse.exe, select default workspace, e.g.



Go to File -> Import. In the import window type ‘maven’ and select ‘Existing maven projects’

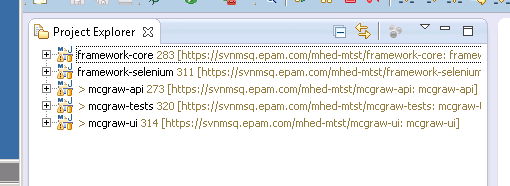
In the Root directory browse and select project folder. In the projects window you will see all the projects



Uncheck the root project and click Finish.

Wait untill Importing process completed. Close Start Screen.

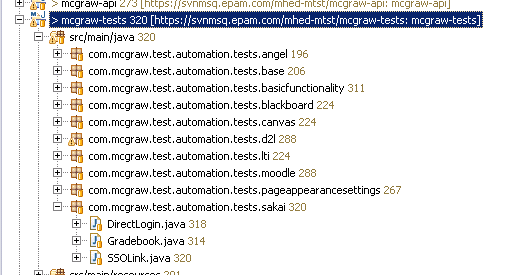
Once projects imported you may find them into Project explorer (on the left side).



We need mcgraw-tests project

Expand mcgaw-tests -> src/main/java

You will see the test classes under packages, e.g. Sakai test classes



Before running tests make sure, that <project folder>\mcgraw-tests\src\main\resources\ base.properties has correct data:

The local run pathes should be uncommented and server run pathes should be commented:

#Rampart path for server run

#blackboard.api.rampart.path=..\\..\\..\\dependencies\\axis2-1.5.6\\repository\\

#Rampart path for local run

blackboard.api.rampart.path=..\\dependencies\\axis2-1.5.6\\repository\\

#Path for server run

#tegrity.deleteinstance.path=..\\..\\..\\dependencies\\DeleteInstance\\DeleteInstance.exe

#Path for local run

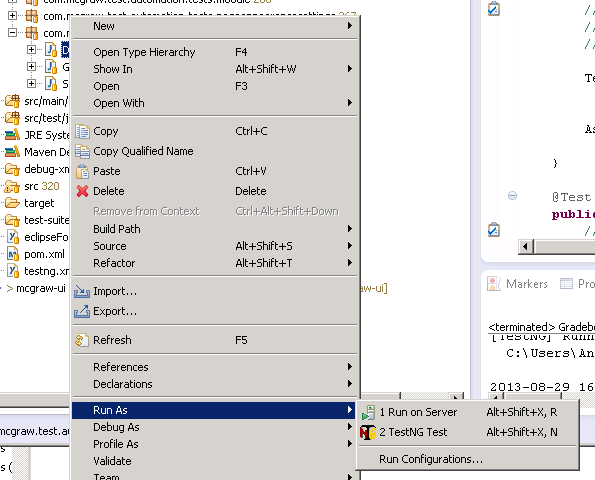
tegrity.deleteinstance.path=..\\dependencies\\DeleteInstance\\DeleteInstance.exe

In the Project Exlplorer Select All files and press F5 to refresh with the changes

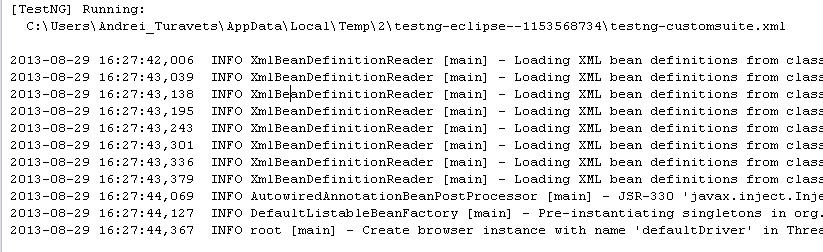
* Running tests

1. The simpliest way to run tests from Eclipse ide, but in this case we don’t get html report(to get test run with completed report see the 2nd way)

To run e.g. DirectLogin.java, right-click on it -> Run As -> TestNG Test

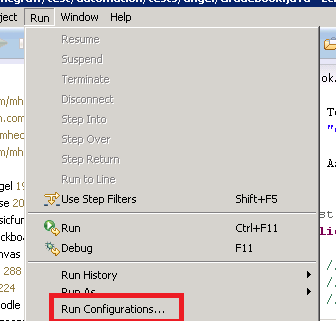


In console view we can see the log of current test:



1. To run tests with Eclipse ide and get full report proceed to next steps:

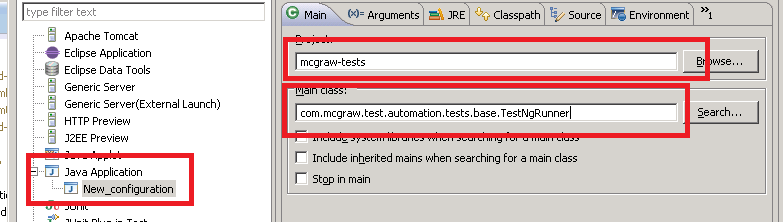
* In top bar click Run -> Run Configurations



* Select Java Application -> New\_Configuration

In the project field enter ‘mcgraw-tests’ if empty

In the Main class field enter ‘com.mcgraw.test.automation.tests.base.TestNgRunner’

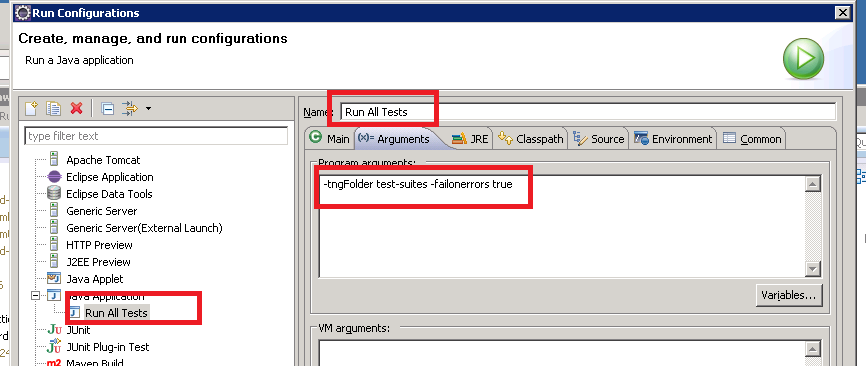


* Go to arguments tab and fill Program arguments:

To run all tests enter: -tngFolder test-suites -failonerrors true

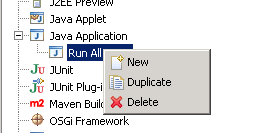
Rename config e.g. to Run All Tests and click apply.

You will get something like this (See screen below):



Now if you click ‘Run’ you will run all the tests.

To create such runners for suites, right click -> on the ‘Run All Tests’ in the left tree and click duplicate



Go to ‘Arguments’ tab and change Program arguments which are should be use by the suite. Programm arguments for the each suite are below:

* Angel Tests: -tng test-suites/Angel.xml -failonerrors true
* BlackBoard Tests: -tng test-suites/Blackboard.xml -failonerrors true
* Canvas Tests: -tng test-suites/Canvas.xml -failonerrors true
* D2L Tests Version 10(default): -tng test-suites/D2L.xml -failonerrors true
* D2L Tests Version 9: -tng test-suites/D2L.xml -failonerrors true -d2lver v9
* Ecollege tests: -tng test-suites/ECollege.xml -failonerrors true
* LTI Tests: -tng test-suites/LTI.xml -failonerrors true
* MH Campus Basic Functionality Tests: -tng test-suites/CampusBasicFunctionality.xml -failonerrors true
* MH Campus External Tools: -tng test-suites/CampusExternalTools.xml -failonerrors true
* MH Campus Page Appearance Tests: -tng test-suites/CampusPageAppearanceSettings.xml -failonerrors true
* Moodle Tests: -tng test-suites/Moodle.xml -failonerrors true
* Sakai Tests: -tng test-suites/Sakai.xml -failonerrors true

The test report will be located here <project-folder>\mcgraw-tests\test-output\html\index.html

This is your report (Note: after test run the old report will be deleted and report for current test run is created)