Tutorial for Basic Jenkins Integration

OSMO Tester

MBT tool

v2.3

Teemu Kanstrén

Table of Contents

[Introduction 3](#_Toc334520287)

[Custom Integration 3](#_Toc334520288)

[Built-In Integration 3](#_Toc334520289)

[Conclusions 4](#_Toc334520290)

[References 4](#_Toc334520291)

# Introduction

This tutorial discusses integrating OSMO Tester test generation with Jenkins. The reader should be familiar with the basic use of OSMO Tester from other tutorials. The reader should also understand their Jenkins. If you are unfamiliar, ask the butler.

# Custom Integration

You can obviously integrate OSMO Tester test generation with Jenkins by your own custom integration solution. In this case, you would have OSMO Tester generate the test cases in one step, store these generated tests in one location, run another task after this to execute the tests and so on. However, in some cases, especially when you are running online tests, you may wish to directly integrate the OSMO Tester results with the Jenkins reporting system. A typical approach for test results reporting in Java based applications in Jenkins is to use the Ant task format. This is what OSMO Tester also supports out of the box. Anything else, the customization is up to you (or someone you pay..).

# Built-In Integration

To generate a Jenkins report, you need to add a specific listener to the OSMO Tester generator. An example of this is shown in Listing 1.

public class OnlineMain {

public static void main(String[] args) {

OSMOTester tester = new OSMOTester(new OnlineHelloModel());

tester.setSeed(345);

OSMOConfiguration config = tester.getConfig();

**JenkinsReportGenerator reporter =**

**new JenkinsReportGenerator("hello-steps.xml", true);**

**config.addListener(reporter);**

config.addTestEndCondition(new Length(5));

config.addSuiteEndCondition(new Length(3));

config.setUnwrapExceptions(true);

tester.generate();

}

}

Listing 1. A Java program to run the build and produce the test results.

When you have added this listener to the generator, it will automatically write a test report for all the executed test cases in the Ant JUnit test report format to the given file after test generation has finished. You then need to configure your Jenkins job to use this report. Once this file has been generated, you can have Jenkins publish it by simply enabling “Publish JUnit test results report” in your build configuration, and giving it the location for the file. The location should define where your generator writes it, which depends on your build configuration.

The parameter “true” that is given to the Jenkins report in Listing 1 defines whether you wish to have all test steps represented as separate test cases in the test report or if you wish to have each generated test case reported as a test case in Jenkins. Having test steps reported as test cases makes it easier sometimes in online testing to see from the report directly how test execution progressed. This can be especially useful in figuring out what happened if there is a failure.

xx shows a brief portion of the Ant build.xml file used to run this example case from Jenkins. The complete build.xml file can be found in the OSMO Tester source code repository under “examples”.

<target name="calendar-jenkins" depends="jar">

<java classname="osmo.tester.examples.report.CalendarJenkins"

classpathref="class.path"/>

</target>

Listing 2. Ant script for running Listing 1.

Following are a few screenshots of Jenkins configuration for the example.

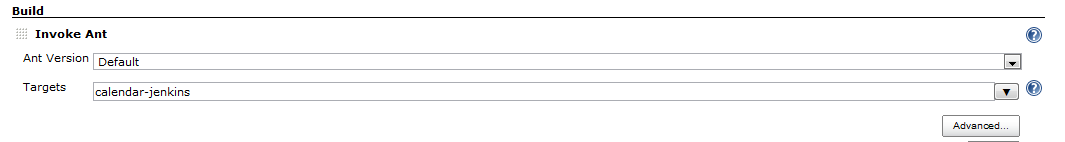


Figure 1. Ant configuration in Jenkins for Listing 2.

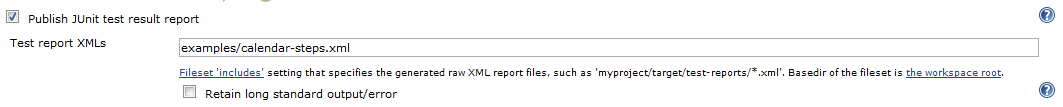


Figure 2. JUnit test report configuration in Jenkins for listing Listing 1.

Notice that in Figure 1, the Ant task name is the same as the name given in Listing 2. And the name of the xml file given in Figure 2 is the same as given in Listing 1. The directory structure would depend on your build setup and configuration.

# Conclusions

This is a short tutorial on Jenkins integration using the OSMO components for Jenkins integration. The approach should be rather simple. For more complex integration, find a Jenkins and OSMO Tester expert.

# References

OSMOTester home page, discussion forums & source code: <http://code.google.com/p/osmo/>