

Summary

In Test Automation, reporting mechanism is important for viewing test results via data processed from txt, XML files. This guide will discuss how to create HTML or Excel reports based on text or XML files after test runs.

Description

Ant and Maven build tools can also be used for generating meaningful test result reports.

However, Ant Task plug-in and Maven plug-in is required for it - for one, HTML plug-ins are open-source, and are called JUnit Report Task, and maven-surefire-report-plugin libraries.

Also, the Excel counterparts are added by the eGovFrame IDE, and they are called Ant Task and Maven plug-in.

Following is how to create reports.

1. Create a script (Ant - build.xml, Maven - pom.xml)
2. Create a report using Eclipse plug-in
3. Confirm report

Creating reports is similar to running tests. You just need to set the goal or target in the CI server to generate reports.

Reports will look as below.

Maven's HTML report

- Ant's HTML report
- Excel report
-

Environmental settings

- Install Ant, Maven Eclipse plug-ins
- Add library to create TestSuite (pom.xml) : refer to How to modify pom.xml dependency for unit tests
- For Excel reporting, modify setting.xml

Manual

Test reports can be generated as .txt, .html, .xml, .xls formats.

Test reports text and XML reports merely hold hold test data instead of real reports, it usually gets generated during test runtime. Therefore, refer to [Running Tests](#) for those cases, and here we will focus on HTML and XML reports.

Creating HTML reports

Creating HTML report using Maven

1. Edit pom.xml : refer to [HTML report example](#)
2. Create report in Eclipse : refer to [How to run Maven on Eclipse](#), and enter surefire-report:report for the goal, and confirm BUILD SUCCESSFUL in the Console View.
3. Confirm HTML report : double-click on target/site/surefire-report.html reports that get created by default.
They should look identical to the Maven reports.

Creating HTML report using Ant

1. Edit build.xml : refer to [HTML report example](#)
2. Create report in Eclipse : refer to [How to run Ant on Eclipse](#), run the target, and confirm BUILD SUCCESSFUL in the Console View.

3. HTML report : in build.xml, you can set todir="`\${testhtml.dir}`"(<report format="frames" todir="`\${testhtml.dir}`"/>), and you can double-click on the index.html directory set as testhtml.dir to confirm. It should be identical to the Ant's HTML report.

Creating Excel report

Excel report creating feature is exclusively developed just for the eGovFrame IDE. This guide will explain how to use default settings to create Excel reports. If you use a custom template Excel file, you can customize the resulting Excel report.

Creating Excel report in Maven

1. setting.xml : you need appropriate settings for using the goal.
2. pom.xml :refer to [Creating Excel report sample](#)
3. Create report in Eclipse : refer to Running Maven in Eclipse, and add egovtest:junit-xls to the goal, and then check BUILD SUCCESSFUL in the Console View.
4. Check Excel report : the default path is target/egovtest/egovtest-junit.xls, so double-click on the file to open and check the contents. template-junit.xls is the default template Excel file.

Creating Excel report in Ant

1. build.xml : refer to [Creating Excel report sample](#)
2. Create report in Eclipse : refer to Running Ant in Eclipse, and run the target, and then check BUILD SUCCESSFUL in the Console View.
3. Check Excel report : in build.xml, you can set todir="`\${testxls.dir}`"(<egov-junitreport todir="`\${testxls.dir}`"/>), and by default, the egovtest-junit.xls file gets generated in the directory specified in the testxls.dir, so double-click the file to check the contents of the report. Also, the template-junit.xls file is the default template Excel file.

Sample

Creating HTML report sample

Maven - pom.xml

```
<reporting>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-report-plugin</artifactId>
      <version>2.4.2</version>
    </plugin>
  </plugins>
</reporting>
```

Ant - build.xml

```
<!-- Junit Test Result Report -->
<target name="junitreport" depends="tests">
  <junitreport todir="${testhtml.dir}">
    <fileset dir="${testreports.dir}">
```

```

        <include name="TEST-*.xml" />
    </fileset>
    <report format="frames" todir="${testhtml.dir}" />
</junitreport>
</target>

```

Creating Excel report sample

■ Maven - pom.xml

1. Default settings

```

<build>
  <defaultGoal>egovtest:junit-xls</defaultGoal>
  <plugins>
    <!-- Egovframework JUnit Excel Reporting -->
    <plugin>
      <groupId>egovframework.dev</groupId>
      <artifactId>egovtest-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>

```

2. All settings

```

<build>
  <defaultGoal>egovtest:junit-xls</defaultGoal>
  <plugins>
    <!-- Egovframework JUnit Excel Reporting -->
    <plugin>
      <groupId>egovframework.dev</groupId>
      <artifactId>egovtest-maven-plugin</artifactId>
      <configuration>
        <outputDirectory>${project.build.directory}/egovtest</outputDirectory>
        <reportsDirectory>${project.build.directory}/surefire-reports</reportsDirectory>
        <outputName>egovtest-report.xls</outputName>
        <templatePath>${basedir}/src/test/resources/unit/egovtest-excel-test/template-kr.xls</templatePath>
        <headerPosition>
          <summary>0,B,6</summary>
          <package>0,B,11</package>
          <lists>1,A,5</lists>
        </headerPosition>
      </configuration>
    </plugin>
  </plugins>
</build>

```

■ Ant - build.xml

1. Default settings

```

<path id="egov.lib">
  <path refid="master-classpath" />
  <path location="${antlib.dir}/egovtest/egovframework-dev-tst-ant.jar" />
</path>

```

```

<taskdef resource="egovtest.properties" classpathref="egov.lib" />

<!-- JUnit Excel Report -->
<target name="egovtest" depends="tests">
    <egov-junitreport todir="{testxls.dir}">
        <fileset dir="{testreports.dir}" includes="**/TEST-*.xml" />
    </egov-junitreport>
</target>

```

2. All settings

```

<path id="egov.lib">
    <path refid="master-classpath" />
    <path location="{antlib.dir}/egovtest/egovframework-dev-tst-ant.jar" />
</path>

<taskdef resource="egovtest.properties" classpathref="egov.lib" />

<!-- JUnit Excel Report -->
<target name="egovtest-full" depends="tests">
    <egov-junitreport todir="{testxls.dir}"
        outputname="egovtest-junit-full.xls"
        templatepath="{basedir}/build/template-kr.xls"
        summary="0,B,6"
        packages="0,B,11"
        lists="1,A,5">
        <fileset dir="{testreports.dir}"
            includes="**/TEST-*.xml" />
    </egov-junitreport>
</target>

```

References

Maven Surefire Plugin Home : <http://maven.apache.org/plugins/maven-surefire-plugin/>
 [http://maven.apache.org/plugins/maven-surefire-plugin/]

Maven Surefire Report Plugin Home : <http://maven.apache.org/plugins/maven-surefire-report-plugin/>
 [http://maven.apache.org/plugins/maven-surefire-report-plugin/]

Ant JUnit Task : <http://ant.apache.org/manual/OptionalTasks/junit.html>
 [http://ant.apache.org/manual/OptionalTasks/junit.html]

Ant JUnit Report Task : <http://ant.apache.org/manual/OptionalTasks/junitreport.html>
 [http://ant.apache.org/manual/OptionalTasks/junitreport.html]

m2eclipse Home : <http://m2eclipse.sonatype.org/index.html> [http://m2eclipse.sonatype.org/index.html]

[Test result Excel report](#)

[Test result Excel report in-depth](#)