

Table of Contents

1 The JUnit Plugin.....	1
1.0.1 Description.....	1
1.0.2 Vendor.....	1
1.0.3 Version.....	1
1.0.4 Homepage.....	1
1.0.5 Documentation.....	1
1.1 Activation.....	1
1.1.1 Suitability.....	1
1.1.2 Execution point.....	1
1.1.3 Profiles.....	2
1.2 Deactivation.....	2
1.3 Input.....	2
1.3.1 Required properties.....	2
1.3.2 Optional properties.....	3
1.3.3 Other requirements.....	4
1.3.4 Dependencies.....	4
1.4 Output.....	4
1.4.1 Module reports.....	4
1.4.2 Global reports.....	4
1.5 Metrics.....	4
1.5.1 Overall.....	4
1.5.2 Module.....	5
1.6 Advanced.....	5
1.6.1 Register.....	5
1.6.2 Execute.....	5
1.6.3 Publish.....	5
1.6.4 Finalise.....	5

1 The JUnit Plugin

1.0.1 Description

JUnit is a unit testing framework for the Java programming language. Created by Kent Beck and Erich Gamma, JUnit is one of the xUnit family of frameworks that originated with Kent Beck's SUnit.

1.0.2 Vendor

JUnit.org is community site based on the JUnit testing framework. The site is hosted and maintained by Object Mentor, a team of highly experienced software professionals specializing in object-oriented technologies and Agile/XP development methodologies.

The JUnit framework is an open-source project hosted on Sourceforge.net, and is not maintained by Object Mentor.

1.0.3 Version

JUnitPlugin 1.5.0-1

JUnit 4.5

1.0.4 Homepage

<http://www.junit.org>

1.0.5 Documentation

JavaDoc: <http://junit.org/junit/javadoc/4.5/>

Getting Started: <http://junit.sourceforge.net/>

1.1 Activation

1.1.1 Suitability

Suitable for any module with available compiled Java source code (class files) or binaries (JAR files) including unit tests.

External libraries needed to run the code must also be available.

1.1.2 Execution point

The execution point defines during which command and which target the plugin will be executed.

COMMAND	Build
TARGET	Pretest

1.1.3 Profiles

The profiles must be specified in the configuration in order to activate the plugin.

The plugin profile can be used to enable specific plugins.

The bundle profiles can be used to enable a set of plugins within a specific category.

PLUGIN PROFILES	junit
BUNDLE PROFILES	java

1.2 Deactivation

If this plugin has been activated for a larger set of modules (at project or subsystem level), it is possible to disable it for specific modules using special properties:

NAME	nojunit
DESCRIPTION	To disable the FindBugs execution for the current module

NAME	notest
DESCRIPTION	To disable all the plugins registered at TEST target (PRETEST, TEST, POSTTEST) for the current module

1.3 Input

1.3.1 Required properties

The required properties must be available in order for the plugin to run.

Some of the requires properties may have default values which correctness must be check to ensure the correct execution of the plugin.

NAME	junit.test.class
DESCRIPTION	Fully qualified name of the Unit Test class to execute. The class must be present in the directory specified by junit.test.location. In case this property is not defined, all the available tests in junit.test.location will be executed
DEFAULT VALUE	All tests in junit.test.location will be executed
EXAMPLE	To execute only the class org.myproject.suites.FastTestSuite, the property must be set as: org.myproject.suites.FastTestSuite

NAME	junit.test.location (java.test.location)
DESCRIPTION	Root of the directory structure where the compiled Java test and mock object files (.class) are placed according to their package definition. java.test.location can be used instead of junit.test.location to have a generic property that works with all the java plugins. All the .class files in directory structure will be added in the CLASSPATH and one or all tests will be executed according to junit.test.class
DEFAULT VALUE	\${src.location}/test/classes or if not present \${src.location}/build/classes-test
EXAMPLE	Assuming the compilation output directory is \${location}/test/class having files like \${src.location}/test/class/org/my/package/MyTest.class, the property must be set as: \${src.location}/test/class

NAME	junit.class.location (java.class.location)
DESCRIPTION	Root of the directory structure where the (non-test) compiled Java files (.class) are placed according to their package definition. Needed only if the classes are needed for the execution of the tests and not already in the CLASSPATH. java.class.location can be used instead of junit.class.location to have a generic property that works with all the java plugins. All the .class files in directory structure will be added in the CLASSPATH.
DEFAULT VALUE	\${src.location}/classes or if not present \${src.location}/build/classes
EXAMPLE	Assuming the compilation output directory is \${location}/build/class having files like \${src.location}/build/class/org/my/package/MyClass.class, the property must be set as: \${src.location}/build/class

NAME	junit.jar.location (java.jar.location)
DESCRIPTION	Directory containing the (non-test) JAR files of the application, if available. Needed only if the JARs are needed for the execution of the tests and not already in the CLASSPATH. java.jar.location can be used instead of junit.jar.location to have a generic property that works with all the java plugins. The available JAR files will be added in the CLASSPATH.
DEFAULT VALUE	\${src.location}/jars or if not present \${src.location}/build/jars
EXAMPLE	Assuming the ANT script generates a JAR file in \${src.location}/build/jars/MyJarFile.jar, the property must be set as: \${src.location}/build/jars

NAME	junit.lib.location (java.lib.location)
DESCRIPTION	Directory containing the external libraries (JAR files) required to execute the tests. Needed only if the JARs are needed for the execution of the tests and not already in the CLASSPATH. java.lib.location can be used instead of junit.lib.location to have a generic property that works with all the java plugins. The available JAR files will be added in the CLASSPATH.
DEFAULT VALUE	\${src.location}/lib or if not present \${src.location}/endorsed or again \${stageDir}/share/java
EXAMPLE	Assuming all the external libraries are stored in \${src.location}/libs/MyExternalDependency.jar, the property must be set as: \${src.location}/libs

NAME	junit.failure
DESCRIPTION	Defines whether a module build must fail in case one or more test fail: true - The module build is marked as failed ; false - No module build is marked as failed due to JUnit
DEFAULT VALUE	false
EXAMPLE	To force module builds to fail in case one or more tests fail, the property must be set as: true

1.3.2 Optional properties

The optional properties can be used to fine tune the execution of the plugin.

NAME	junit.jvmargs
DESCRIPTION	Arguments to be passed to the Java virtual machine used to run the tests. Each test will be run in a separate virtual machine
DEFAULT VALUE	
EXAMPLE	To allocate a system property for the tests like key=value, the property must be set as: -Dkey=value

1.3.1 Required properties

NAME	junit.timeout
DESCRIPTION	Amount of time, in milliseconds, that a single test may run before assuming it to be hung and cancel it. In case the property is not defined, JUnit will wait as long as needed for the test to complete.
DEFAULT VALUE	600000 (10 minutes)
EXAMPLE	To set the timeout to 20 minutes, the property must be set as: 1200000

1.3.3 Other requirements

NONE

1.3.4 Dependencies

NONE

1.4 Output

1.4.1 Module reports

For each module, the output is generated in `${src.location}/reports/junit`

The plugins generate a JUnit XML file: `TEST-TestSuites.xml` and it uses the default JUnit XSL style sheets to generate the default html reporting accessible via `index.html`

To check the XML schema and style sheet, please refer to the JUnit original documentation.

1.4.2 Global reports

The final summary is generated in `${workspaceDir}/reports/junit`

Each generated JUnit XML and HTML files are also available in the global reports directory under `${workspaceDir}/reports/junit/${moduleName}-${name}/TEST-TestSuites.xml`, `${workspaceDir}/reports/junit/${moduleName}-${name}/index.html` and all the other HTML files.

1.5 Metrics

1.5.1 Overall

NAME	JUnit
VALUE	Total number of tests successful executed
UNIT	Test
TYPE	Integer
CONTEXT	
LINKS	HTML: <code>\${workspaceDir}/reports/junit/index.html</code>
VALUES	tests : the number of successful tests ; errors : the number of errors ; failures : the number failures

NAME	JUnit rate
VALUE	Percentage of components with successful tests executed
UNIT	Module
TYPE	Percentage
CONTEXT	

LINKS	HTML: <code>\${workspaceDir}/reports/junit/index.html</code>
VALUES	success : the number of modules with successful tests ; errors : the number of modules with tests executing with errors ; failures : the number of modules with tests executing with failures ; notest : the number of modules without tests

1.5.2 Module

The module metrics are related to a single module and provide its plugin execution details.

NAME	JUnit
VALUE	Number of tests successfully executed
UNIT	Test
TYPE	Integer
CONTEXT	
LINKS	HTML: <code>\${workspaceDir}/reports/junit/index.html</code> ; XML: <code>\${workspaceDir}/reports/junit/\${moduleName}-\${name}/TEST-TestSuites.xml</code>
VALUES	tests : the number of tests successfully executed ; errors : the number of errors ; failures : the number of failures

1.6 Advanced

1.6.1 Register

Verifies dependencies and the existence of the ANT build file.

Registers the plugin

1.6.2 Execute

Verifies the values of user defined properties The main functionality is executed using build-junit.xml running 'run' target.

1.6.3 Publish

Detailed HTML reports are generated and module metrics are added to build-status.xml.

Link to the generated HTML report is added to the build-status.xml

1.6.4 Finalise

Detailed HTML summary reports is generated and overall metrics are added to build-status.xml. Link to the generated HTML report is added to the build-status.xml -- FabioCapannini - 16-Sep-2011

This topic: EMI > EticsPluginsJUnit

Topic revision: r2 - 2011-09-19 - FabioCapanniniExternal



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback