

Multi Module Tests Publisher

Plugin Information

View multi-module-tests-publisher [on the plugin site](#) for more information.

This plugin stores junit results into an embedded derby database and groups testcases by module,package,class and case


This plugin can be used as a replacement for the built-in junit plugin.


This plugin allows junit tests to be grouped by suites in addition to the being grouped by package, class and case.


It persists the test into a derby database that runs in embedded mode. This is done so that reports can be generated easily using sql queries. This also reduces the memory required to display reports and history

The trend graph for each suite is displayed on the project home page. Three new columns, Success Rate, Failure Rate and Skip Rate are available that displays the percentage rate of success, failure and skip at the project level, suite level, package level, test class level and case level

Project TestJUnitSuite

 [Workspace](#)

 [Recent Changes](#)

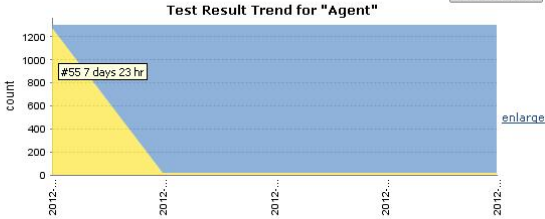
 [Latest Test Result \(no failures\)](#)

Permalinks

- [Last build \(#59\), 20 hr ago](#)
- [Last stable build \(#59\), 20 hr ago](#)
- [Last successful build \(#59\), 20 hr ago](#)

[add description](#)

Disable Project

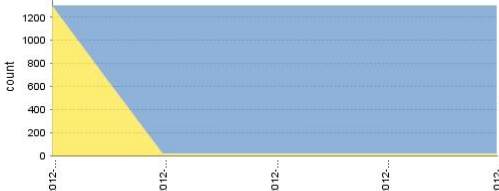


Test Result Trend for "Agent"

count

#55 7 days 23 hr

enlarge



Test Result Trend for "Core"

count

(just show failures)

Test Report

Overall Results

0 failures (±0) , 18 skipped (±0)

2,601 tests (±0)

(no description)

[edit description](#)

Agent

0 failures (±0) , 9 skipped (±0)

1,301 tests (±0)

Core

0 failures (±0) , 9 skipped (±0)

1,300 tests (±0)

Result Breakdown by Group

	Pass	(diff)	Fail	(diff)	Skip	(diff)	Total	(diff)	Success Rate	Failure Rate	Skip Rate
Agent	1292		0		9		1301		79.82%	0.00%	20.18%
Core	1291		0		9		1300		79.46%	0.00%	20.54%
Total	2583		0		18		2601		79.64%	0.00%	20.36%

The screenshot of the configuration page is shown below

[Cancel](#)

Post-build Actions

- Publish JUnit test result report ?
- Build other projects ?
- Record fingerprints of files to track usage ?
- Publish Javadoc ?
- Archive the artifacts ?
- Aggregate downstream test results ?
- Publish JUnit test result report grouped by suite

JUnit Report File Mask
- Retain long standard output/error
- E-mail Notification ?