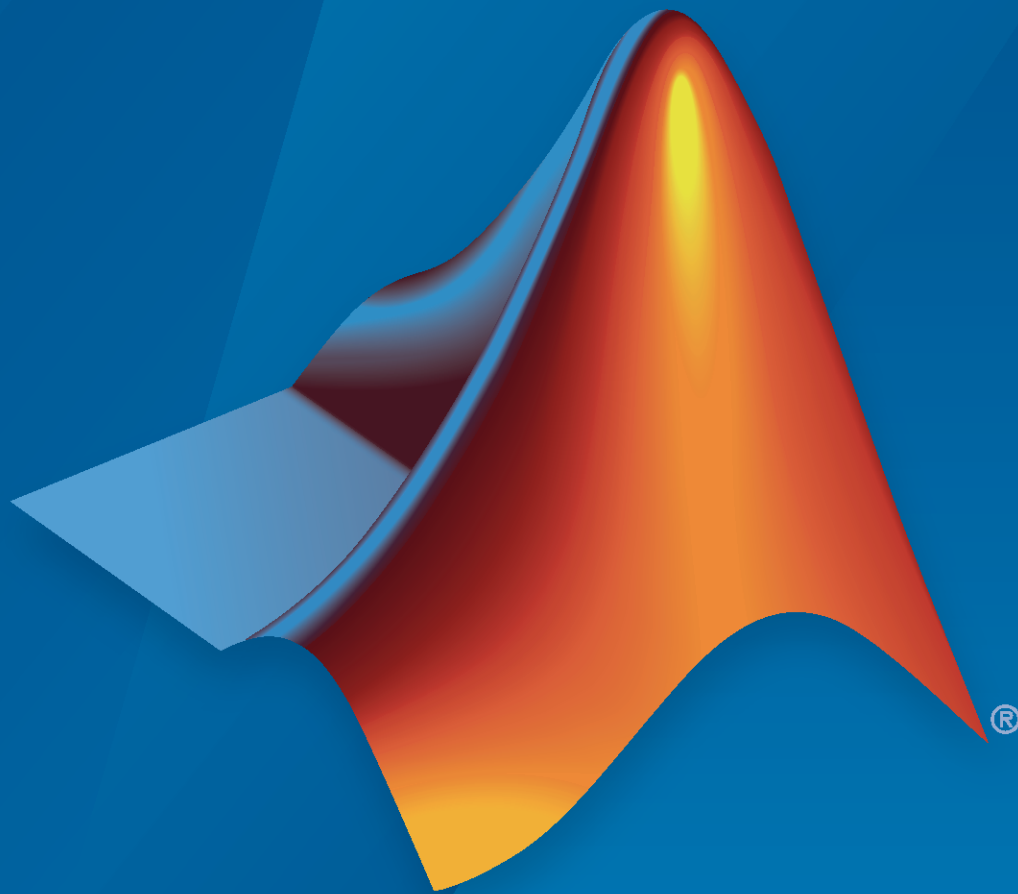


MATLAB® Test™ Release Notes



MATLAB®



How to Contact MathWorks



Latest news: www.mathworks.com
Sales and services: www.mathworks.com/sales_and_services
User community: www.mathworks.com/matlabcentral
Technical support: www.mathworks.com/support/contact_us



Phone: 508-647-7000



The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

MATLAB® Test™ Release Notes

© COPYRIGHT 2023 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

R2023a

Measure and monitor MATLAB code quality for projects by using the Code Quality Dashboard	1-2
Manage MATLAB tests and results for projects by using MATLAB Test Manager	1-2
Measure the decision, condition, and modified condition/decision coverage (MC/DC) of your source code	1-2
Generate C/C++ code using MATLAB Coder and test for equivalence with MATLAB source code	1-3
Generate deployed code artifacts using MATLAB Compiler SDK and test for equivalence with MATLAB source code	1-3
Filter test suites by source code dependency	1-3
Create test suites from tests that verify requirements	1-4
Verify requirements by running tests in the MATLAB Test Manager	1-4
Run tests using "n-wise" parameter combination	1-4

R2023a

Version: 1.0

New Features

Measure and monitor MATLAB code quality for projects by using the Code Quality Dashboard

Measure and monitor the quality of MATLAB® code and tests in projects by using the **Code Quality Dashboard**. The dashboard collects and displays metrics for MATLAB code and test files (.m files) and requirement sets, including code analysis, test results, coverage results, and requirements verification.

For more information, see “Measure and Monitor Project Quality”

Manage MATLAB tests and results for projects by using MATLAB Test Manager

Manage MATLAB tests and test results for projects by using the **MATLAB Test Manager**. You can:

- Organize tests by creating test suites and using test tags.
- Run tests and view a summary of the results.
- Investigate test results by viewing failure diagnostics and using filters.
- Run subsets of tests by using test suites, test selections, and filters.

For more information, see “Manage Tests and Results by Using the MATLAB Test Manager” and **MATLAB Test Manager**.

Measure the decision, condition, and modified condition/decision coverage (MC/DC) of your source code

You can measure the decision coverage, condition coverage, and modified condition/decision coverage (MC/DC) of your source code when you run tests with MATLAB Test™. To perform a code coverage analysis using these coverage types, specify the `MetricLevel` name-value argument when you create a plugin using one of the static methods of the `CodeCoveragePlugin` class. For example, run your tests and generate an HTML code coverage report that includes all supported metrics for the source code in a folder.

```
import matlab.unittest.plugins.CodeCoveragePlugin
import matlab.unittest.plugins.codecoverage.CoverageReport

suite = testsuite("MyTestClass");
runner = testrunner("textoutput");
format = CoverageReport;
plugin = CodeCoveragePlugin.forFolder("myFolder", ...
    Producing=format,MetricLevel="mcdc");
runner.addPlugin(plugin)
results = runner.run(suite);
```

The `MetricLevel` argument specifies which coverage types to include in the analysis. This table shows the possible values of `MetricLevel` and the corresponding coverage types.

Value of <code>MetricLevel</code>	Types of Coverage Included
"statement" (default)	Statement and function coverage
"decision"	Statement, function, and decision coverage

Value of MetricLevel	Types of Coverage Included
"condition"	Statement, function, decision, and condition coverage
"mcdc"	Statement, function, decision, condition, and modified condition/decision coverage

For more information on how to collect decision coverage, condition coverage, and MC/DC metrics, see “Collect Code Coverage Metrics for MATLAB Source Code”.

Generate C/C++ code using MATLAB Coder and test for equivalence with MATLAB source code

You can write equivalence tests that use MATLAB Coder™ to:

- Generate C/C++ code from MATLAB source code.
- Execute the generated code with specified inputs.
- Verify that the execution of the generated code and the MATLAB source code match.

For more information, see “Generate C/C++ Code and Test for Equivalence”.

You can also collect coverage to analyze how your equivalence test executes the generated C/C++ code. For more information, see “Collect Coverage for Generated C/C++ Code in Equivalence Tests”.

Generate deployed code artifacts using MATLAB Compiler SDK and test for equivalence with MATLAB source code

You can author equivalence tests that use MATLAB Compiler SDK™ to:

- Build a Java® or Python® package or .NET assembly.
- Execute the deployed code artifact with specified inputs.
- Verify that the execution of the deployed code artifact and the MATLAB source code match.

For more information, see “Generate Deployed Code Artifacts and Test for Equivalence”.

Filter test suites by source code dependency

Use the `matlabtest.selectors.DependsOn` class to filter a test suite based on source code required by your tests. You can use the class to select tests whose test files depend on specified source files or folders. For example, you can create a test suite and then filter it to include only the tests that depend on a source file:

```
import matlabtest.selectors.DependsOn

suite = testsuite("MyTestClass");
filteredSuite = suite.selectIf(DependsOn(fullfile("mySource", "myFile.m")));
```

You also can filter test suites by specifying the `DependsOn` name-value argument for these methods and functions:

- Static methods of the `TestSuite` class, such as `matlab.unittest.TestSuite.fromFolder`

- `selectIf`
- `testsuite`
- `runtests`
- `runperf`

Create test suites from tests that verify requirements

If you have Requirements Toolbox™ installed, you can use the `matlab.unittest.TestSuite.fromRequirements` method to create a test suite that contains the tests that verify specified requirements. For example, you can create a suite of tests that verify a single requirement:

```
reqSet = slreq.load("myRequirementSet.slreqx");  
req = reqSet.find(ID="R1.1");  
suite = matlab.unittest.TestSuite.fromRequirements(req);
```

You also can use the `testsuite`, `runtests`, and `runperf` functions to create test suites from tests that verify requirement sets or to run such tests. For example, `results = runtests("myRequirementSet.slreqx")` runs the tests that verify the specified requirement set.

Verify requirements by running tests in the MATLAB Test Manager

You can verify requirements in your project by linking them to tests in the project and running the tests using the **MATLAB Test Manager**. You can then view a version of the requirements verification status in the **Code Quality Dashboard** that displays only the results for the linked MATLAB tests run in the **MATLAB Test Manager**.

Test results from the **MATLAB Test Manager** do not affect the verification status in the **Requirements Editor**.

For more information, see “Verify Requirements in Projects with MATLAB Tests”.

Run tests using "n-wise" parameter combination

When you pass more than one parameterization property to a method of a `TestCase` subclass, MATLAB Test can invoke the method for every n -tuple of parameter values. To use this type of parameter combination, specify the `ParameterCombination` attribute of the method as “ n -wise”, where n is an integer between 2 and 10.

You can use the “2-wise” and “pairwise” attribute values interchangeably. The “ n -wise” parameter combination generalizes the “pairwise” combination, and typically results in fewer tests and therefore faster test execution compared with the “exhaustive” combination. For more information about parameter combination strategies, see “Use Parameters in Class-Based Tests”.