

Generating MATLAB Tests from Source Code

Introduction

Testing is an essential part of software development, ensuring that the code functions as expected and meets the desired requirements. MATLAB, a popular programming language and environment, provides a comprehensive testing framework called MATLAB Test that allows developers to create, execute, and manage tests for MATLAB code. In this report, we will explore how to generate MATLAB tests from source code using the MATLAB Test framework.

MATLAB Test Framework

The MATLAB Test framework provides tools for developing, executing, measuring, and managing dynamic tests of MATLAB code, including deployed applications and user-authored toolboxes. It enables developers to organize and manage tests and results in their projects, allowing them to group, save, and run custom test suites at scale. The framework also supports industry-standard code coverage metrics, such as condition, decision, and modified condition/decision coverage (MC/DC), to identify untested code paths and assess testing effectiveness and completeness.

Benefits of MATLAB Test

Using MATLAB Test to generate tests from source code offers several benefits:

1. **Improved Code Quality:** MATLAB Test allows developers to visualize and measure the quality of their MATLAB code using the code quality dashboard. The dashboard provides an aggregated summary view of static code analysis and test results, as well as coverage and requirements metrics.
2. **Reduced Test Execution Time:** MATLAB Test enables developers to optimize productivity by only rerunning select tests to verify code edits. This feature is particularly useful for large test suites, where running all tests can be time-consuming. Additionally, developers can leverage dependency-based test selection, run tests in parallel, or integrate tests within continuous integration systems to further reduce test execution time.
3. **Compliance with Industry Standards:** MATLAB Test provides support for industry standards through the IEC Certification Kit. This kit enables developers to qualify MATLAB Test and other MathWorks code generation and verification tools to ISO 26262, IEC 61508, EN 50128, ISO 25119, and related functional safety standards such as IEC 62304 and EN 50657.
4. **Functional Equivalence Testing:** MATLAB Test allows developers to verify the expected behavior of artifacts created with MATLAB Coder and MATLAB Compiler SDK using functional equivalence testing techniques. This ensures that the generated code behaves identically to the original MATLAB code.

Generating MATLAB Tests from Source Code

To generate MATLAB tests from source code using the MATLAB Test framework, developers can follow these steps:

1. **Create a Test File:** In MATLAB, create a new test file using the MATLAB Test framework. This file will contain the test cases for the source code.
2. **Write Test Cases:** Within the test file, write test cases that exercise different aspects of the source code. Test cases should cover various input scenarios and expected output values to ensure comprehensive test coverage.
3. **Execute Tests:** Run the test file using the MATLAB Test framework to execute the test cases against the source code. The framework will provide detailed feedback on the test results, including any failures or errors encountered during the execution.
4. **Analyze Test Results:** Analyze the test results to identify any failures or errors in the source code. The MATLAB Test framework provides tools to visualize and measure code quality, including code coverage metrics, to assess the effectiveness and completeness of the tests.
5. **Refine and Iterate:** Based on the test results, refine the source code to address any failures or errors. Repeat the process of writing test cases, executing tests, and analyzing results until the source code meets the desired requirements and passes all tests.

Example: Testing a MATLAB Algorithm

Let's consider an example of how to use MATLAB Test to verify a MATLAB algorithm. Suppose we have a MATLAB function called `myAlgorithm` that takes an input vector and performs some computations. We want to generate tests for this algorithm to ensure its correctness.

1. **Create a Test File:** In MATLAB, create a new test file called `myAlgorithmTest.m` using the MATLAB Test framework.
2. **Write Test Cases:** Within the `myAlgorithmTest.m` file, write test cases that cover different scenarios for the `myAlgorithm` function. For example, we can write test cases to verify the algorithm's behavior with different input vectors, including edge cases and boundary values.

```
``matlab classdef myAlgorithmTest < matlab.unittest.TestCase methods (Test) function
testAlgorithmWithEmptyInput(testCase) input = []; expectedOutput = []; actualOutput =
myAlgorithm(input); testCase.verifyEqual(actualOutput, expectedOutput); end
```

```
    function testAlgorithmWithPositiveInput(testCase)
        input = [1, 2, 3];
        expectedOutput = [2, 4, 6];
        actualOutput = myAlgorithm(input);
        testCase.verifyEqual(actualOutput, expectedOutput);
    end

    % Add more test cases as needed
end

end ``
```

3. **Execute Tests:** Run the `myAlgorithmTest.m` file using the MATLAB Test framework to execute the test cases against the `myAlgorithm` function.
4. **Analyze Test Results:** Analyze the test results to identify any failures or errors in the `myAlgorithm` function. The MATLAB Test framework will provide detailed feedback on the test results, including any failures or errors encountered during the execution.
5. **Refine and Iterate:** Based on the test results, refine the `myAlgorithm` function to address any failures or errors. Repeat the process of writing test cases, executing tests, and analyzing results until the `myAlgorithm` function meets the desired requirements and passes all tests.

Conclusion

Generating MATLAB tests from source code using the MATLAB Test framework is a powerful technique for ensuring the correctness and quality of MATLAB code. By following the steps outlined in this report, developers can create comprehensive test suites that cover different scenarios and edge cases. The MATLAB Test framework provides tools for executing tests, analyzing results, and visualizing code quality, enabling developers to optimize productivity and meet industry standards.

References:

1. [Getting Started with MATLAB Test](#)
2. [MATLAB Coder Quick Start Guide](#)
3. [MATLAB Test Documentation](#)
4. [MATLAB Unit Test Framework Documentation](#)
5. [Running Tests Using the Test Browser](#)
6. [Executable Program Generation and Verification](#)
7. [MATLAB Test Product Page](#)
8. [Test Case Generation](#)
9. [Generate Test Cases from Model Components](#)
10. [Class-Based Unit Tests](#)
11. [Unit Test Generated Code with MATLAB Coder](#)
12. [Source Code Generation and Verification](#)
13. [Testing Code Generated from MATLAB Code](#)