matlab-unit-testing

Examples and resources to getting started with MATLAB's Unittest Framework.

Getting Started

Definition

What is a 'unit test?' A unit test is a way of testing a unit - the smallest piece of code (e.g., function or program) that can by isolated and tested for correctness, i.e., they work as intended.

"[unit testing] is a software testing method by which individual units of source code - sets of one or more computer modules together with associated control data, usage procedures, and operating procedures - are teste to determine whether they are fit for use." - Wikipedia

For more information, check out this wiki page.

Unit Testing in MATLAB

Unit testing is available through MATLAB Testing Frameworks. There are several ways to write unit tests in MATLAB:

- Script-Based Unit Tests
- Function-Based Unit Tests
- Class-Based Unit Tests

In general, the class-based unit tests are is most advanced and feature-rich way to write unit tests. In order of simple to advanced:

1. script-based
2. function-based
3. class-based

The main advantages of function and class based unit tests are the use of fixtures and testCase qualifications.

For a detailed overview of the different types of MATLAB Unit Tests, see the Ways to Write Unit Tests.

Unit Test Naming

Naming of unit tests should follow the MATLAB-defined conventions for each type of unit test. In general, the following convention can be used for every type of unit test:

1. The file containing the unit tests must start or end with 'test'.
2. The unit test names should start with 'test'.

For specific conventions for *how to name* unit tests, see additional resources.

Unit Testing

1. Write a function/program (`foo.m`)

2. Write a unit test file (`fooTest.m`)
3. Run the unit test(s)
    1. `results = runtests('foo')`
    2. `results = run(foo)`

Writing unit tests can be found here:

- Script-Based Unit Test
- Function-Based Unit Test
- Class-Based Unit Test

Examples of writing unit tests with scripts, functions, and classes are in Examples.

For more advanced topics like fixtures (test setup and teardown), parameterized tests, or creating test suites, see additional resources.

Additional Resources

Unit Test Naming Resources

Additional information on unit test naming styles. Unit tests should convery easy-to-understand information about what is being tested.

- Medium article
- DZone article
- Stack Overflow thread

TestCase Qualifications

Unit test qualifications are verifiable, assumable, or assertable 'checks' used to determine the outcome (i.e., pass/fail) of the unit test.

See more information on all the available unit test qualifications.

Unit Test Fixtures

Test fixtures are used to perform setup and teardown actions before each test or set of unit tests. For example, adding and removing a folder to path to perform the test(s).

- Function-Based Fixtures
- Class-Based Fixtures

Parameterized Tests

Perform unit test(s) with multiple values.

- Write Parameterized Tests
- Write Basic Parameterized Tests