MathWorks, Inc.

MATLAB® Tool Validation Kit

User Guide

MathWorks Consulting
10-11-2022

Tool Validation Kit App Guide

1 About MATLAB® Validation

To comply with the US Food and Drug Administration's Quality System Regulations, any software tool that is used as part of a medical device development process must be validated as fit for its intended use. Such validation of MATLAB, which requires documentation of context of use, risks of failure, and mitigation strategies, is outside the scope of this document. However, a very common risk mitigation strategy for commercial off-the-shelf software is to run a set of validation tests to confirm correct results on the end user's computer.

This document describes how to run such a set of tests from The MathWorks. The resulting test report may be used as evidence for a MATLAB software validation plan. In addition, documentation is provided to extend these tests for any additional features that are considered high risk for the end user's particular context of use.

A test suite is provided for many commonly-used features of MATLAB, Signal Processing Toolbox, Image Processing Toolbox, Statistics and Machine Learning Toolbox, Computer Vision Toolbox, and Deep Learning Toolbox. Major add-on features for additional toolboxes, warrant additional test cases. The test report is organized into chapters based on these large groups of functionalities.

Some typical contexts of use for MATLAB include

- Data analysis
- Algorithm design
- Automating software implementation via production code generation
- Test and measurement applications such as automated test equipment
- Incorporating MATLAB into a medical device

The contents of this kit are oriented toward the data analysis and algorithm design use cases. Production code generation is a higher risk usage, and requires additional verification activities. Please see our IEC 61508 Certification Kit (http://www.mathworks.com/products/iec-61508/) for further recommendations.

The latter two use cases require additional verification and validation activities beyond what is contained in this kit. In particular, using MATLAB as part of a medical device by installing it or deploying an application with MATLAB Compiler requires compliance with FDA's *Guidance for Industry, FDA Reviewers, and Compliance on Off-The-Shelf Software Use in Medical Devices* (1999).

2 Tool Validation Kit Installation

2.1 Required Software

To utilize the Tool Validation Planning Kit, MATLAB and the toolboxes to be validated should be installed. In addition, a valid license for MATLAB, all toolboxes to be validated, and MATLAB Report Generator is needed. It is possible to run the Tool Validation Kit without the MATLAB Report Generator, however it will be more difficult to format the results for the report.

To run the App, release R2022a or later must be installed.

2.2 INSTALLATION FROM **MLTBX** FILE

Navigate to your downloads folder, or wherever **Tool Validation Kit.mltbx** is, and double-click the file.

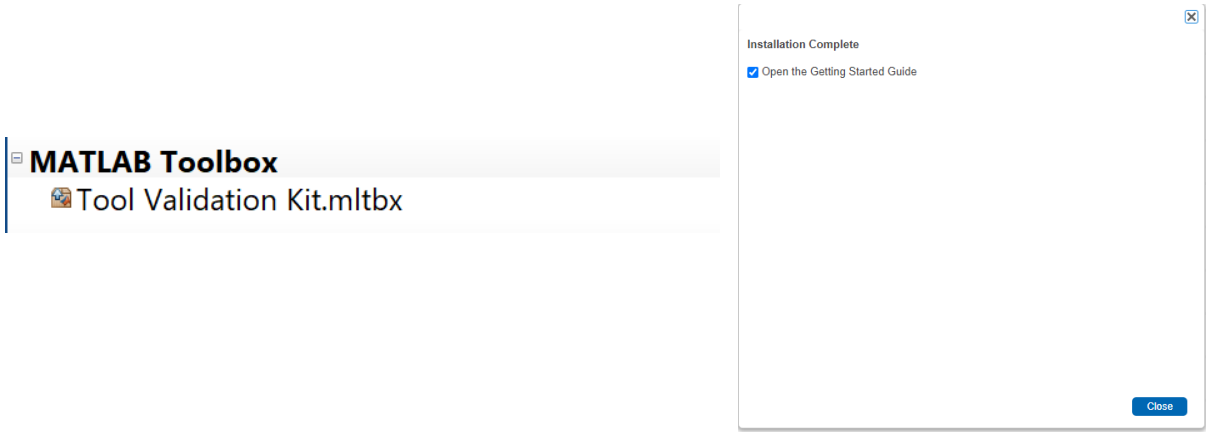

Figure 1. Installing the Tool Validation Kit from an mltbx file.

An installation pop-up window will open prompting you to install the toolbox. During installation of the toolbox, the app will also be installed and will show up in the app gallery. If you already have an installation of the Tool Validation Kit toolbox, it will notify you and ask if you would like to reinstall.

2.3 UNINSTALLING THE TOOL VALIDATION KIT

To uninstall the Tool Validation Kit, navigate to the Apps tab in the toolstrip and right click on the *Tool Validation Kit* App. Select "*Uninstall"* from the options and click "*OK"* in the confirmation pop up. To uninstall the entire toolbox, including the app, navigate to the "Manage Add-Ons", right-click on Tool Validation Kit, and select "Uninstall".

3 BASIC LAYOUT AND USAGE

3.1 USING THE APP

3.1.1 Running for the first time

The MATLAB® Tool Validation Kit App comes with a prewritten collection of test cases in a zip file in the installation folder. Please unzip the UnitTestFolder.zip into any location on disk. On its first run, select the "Browse" button in the "Test Folder" panel to select the location where you have saved the collection of prewritten tests.

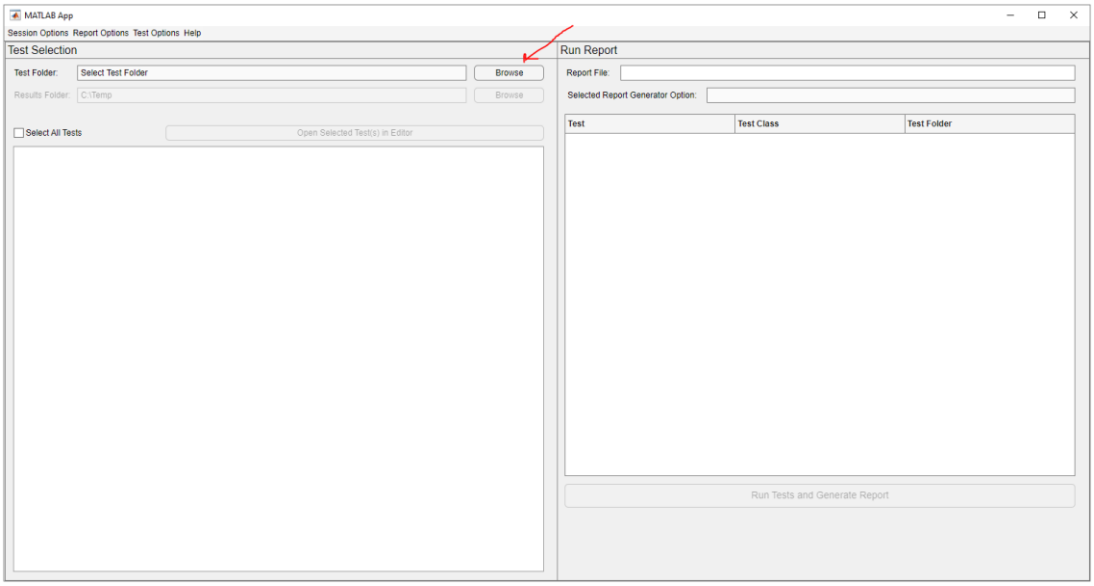

Figure 2. First launch of the Tool Validation App unpacks the provided test cases into a user specified directory.

This will become the default folder the app opens to. The default test folder can be changed by selecting the Browse button on the Test Selection panel.

3.1.2 Run Tests and Generate Report

The "Run Tests and Generate Report" button is initially disabled. To run the Tool Validation Kit, select which tests need to be run to enable the run button.

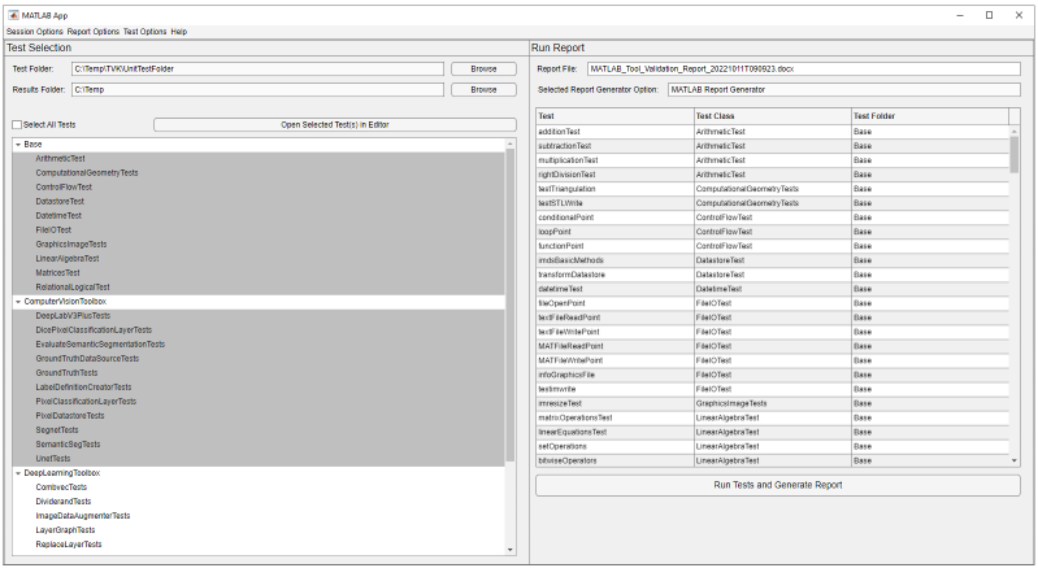

Figure 3. Selecting and running test cases.

The test progress will be displayed at the MATLAB® Command Window and the completed report will then be automatically opened for viewing and editing.

3.1.3 Specify the Report Output Folder

To save the generated report to a specific folder, select the "Browse" button in the Results Folder panel. The default location will be the current folder when launching the app. To customize the report file name, change the value in the "Report File" panel in the "Run Report" section. The default name includes the current datetime.

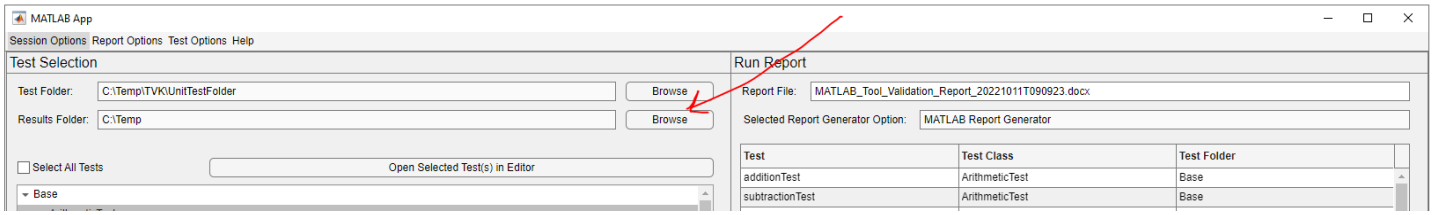

Figure 4. Specifying a default report output folder.

3.1.4 View and Edit the Test Cases

The individual test cases are all editable and can be accessed through the app. Select the test(s) you wish to open and click the "Open Selected Test(s) in Editor" button.

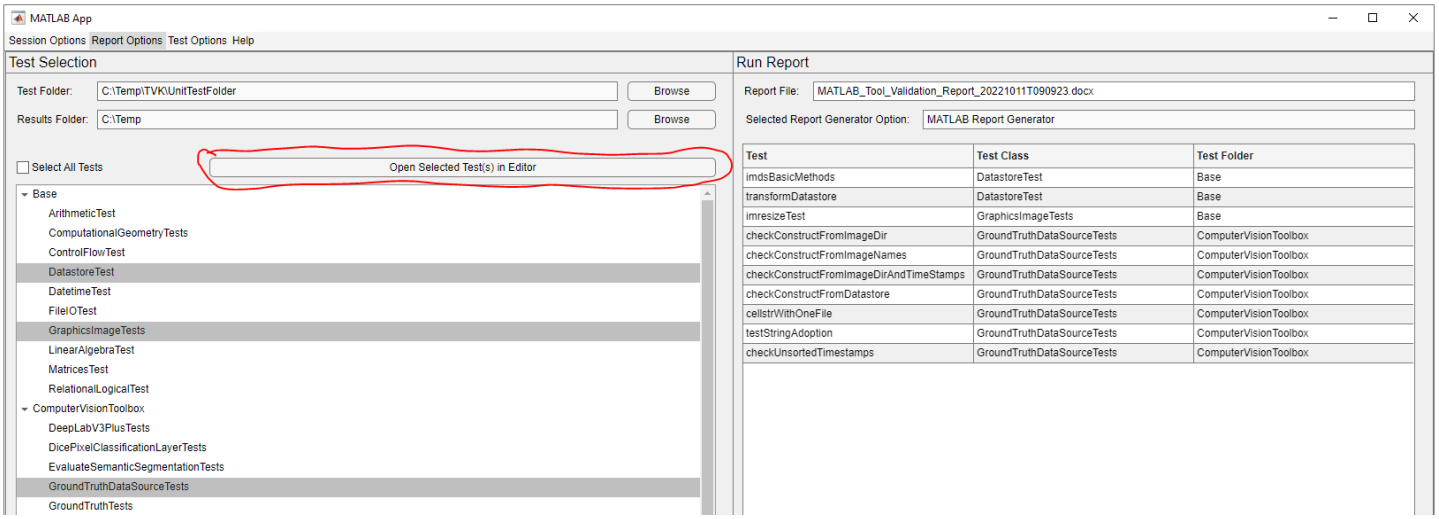

Figure 5. Editing existing test cases.

3.1.5 Selecting the Reporting Option

Through the app, there are two report generation options to choose from, using the MATLAB® Report Generator, or the Test Report Plugin in core MATLAB®. If the MATLAB® Report Generator is not installed, the app will default to the `TestReportPlugin` included in the unit testing framework as of R2016b. To change and save the reporting option, select your preferred reporting option from the "Report Options" menu.

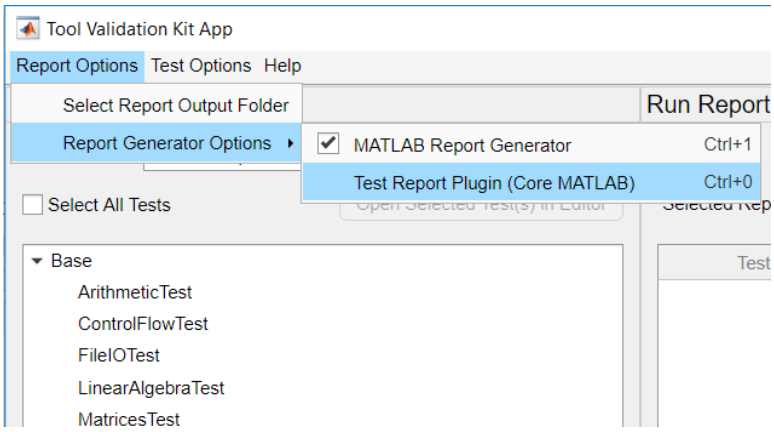

Figure 6. Selecting a reporting option.

3.1.6 Customizing the Tool Validation Report using the Reporter API

The Tool Validation Kit comes with a function that uses the MATLAB Report Generator's Reporter API to build the Tool Validation Report in Microsoft® Word. This function can be customized to fit with your organization's Tool Validation plan and templates.

To make simple changes to Chapter 1 of the report, edit the text files that contain the various sections (Objectives, Scope, etc) in the folder labeled "IntroductionFiles". This is typically located in the App installation folder ('`userpath/Add-Ons/Apps/ToolValidationKitApp`'). When adding sections to the first chapter of the report, maintain the naming convention of "Intro#_" in front of the section name. Save new sections as a text file.

To learn more about making more advanced changes to the Reporter API, please search documentation for the `mlreportgen.report.Reporter` class.

3.1.7 Save and Load a Session

In order to save a session, select the "Save Session" option under "Session Options" in the app toolbar. This will allow you to save a MAT file with all metadata stored for the current app session.

To re-load a previously saved session, simply select "Open Session" option under "Session Options" in the app toolbar, and select the saved MAT file.

4 EXTENDING THE TOOL VALIDATION KIT USING `TEMPLATETEST.M`

A template test case for creating your own validation tests is included in the App installation folder. It can be accessed directly from the App by selecting the "Test Options" menu and clicking on the "Create New Test" option.

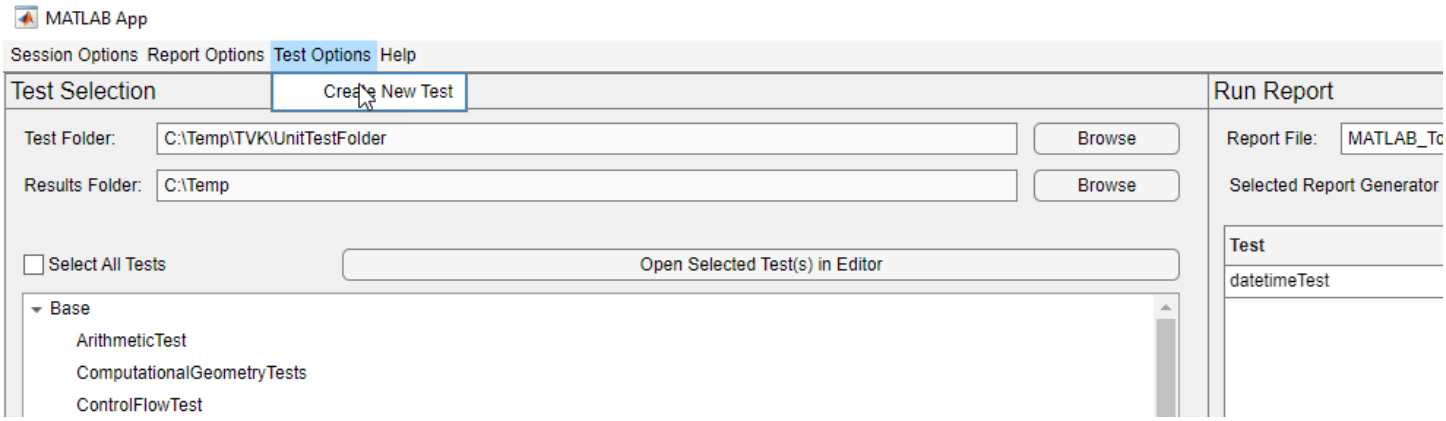

Figure 7 Create a new test

Creating a new test case will copy the new test to a file of your choosing, rename the file to what the user specifies, and open the template in the editor window. The only major change you need to make to this file are the functions defined in the "Test" methods. These must be edited to run the tests you want to run. The other methods can be modified by may change the behavior of the tests on execution. The app should automatically refresh the test list with the new test, however if it doesn't, please re-select the test folder in the app.

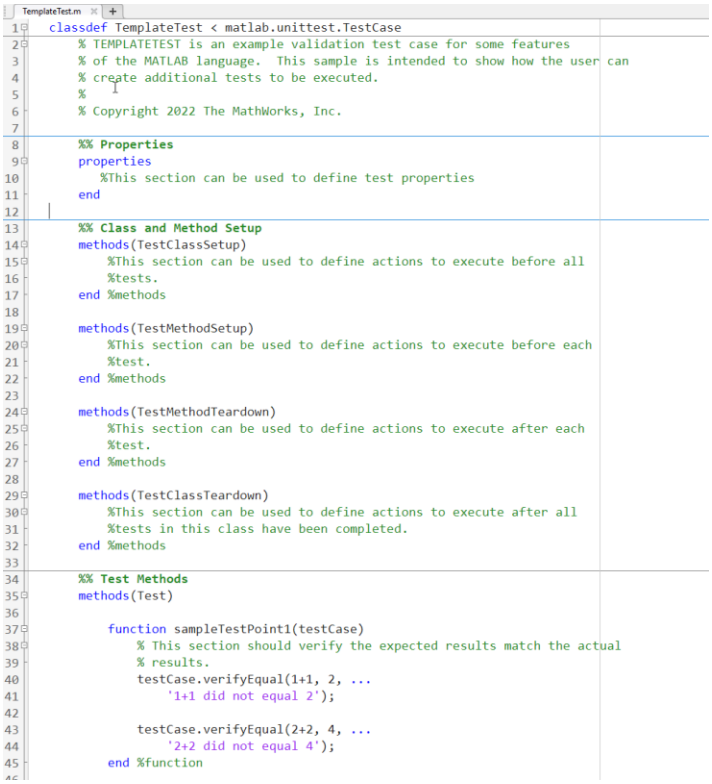

Figure 8 Changing the template test