

MATLAB/Simulink Model Testing

Abstract

TESSY provides MATLAB/Simulink model tests in the same convenient way as you would do normal unit tests. Based on the interface of code generated from the model, test can be defined and executed using the MATLAB tooling.

Table of contents

Abstract	1
1 Introduction.....	2
2 TESSY Environment Settings.....	3
3 dSPACE TargetLink	5
4 Interactive Debugging.....	6
5 Troubleshooting.....	6

1 Introduction

The MATLAB model test adaptation provides a way to test normal MATLAB/Simulink models as well as TargetLink Models. In non-interactive mode MATLAB is started in batch mode without showing the UI. In interactive mode the MATLAB IDE is started and the script runs and stops right before the ports are filled with the test data so that you can check the test data and step through your MATLAB model.

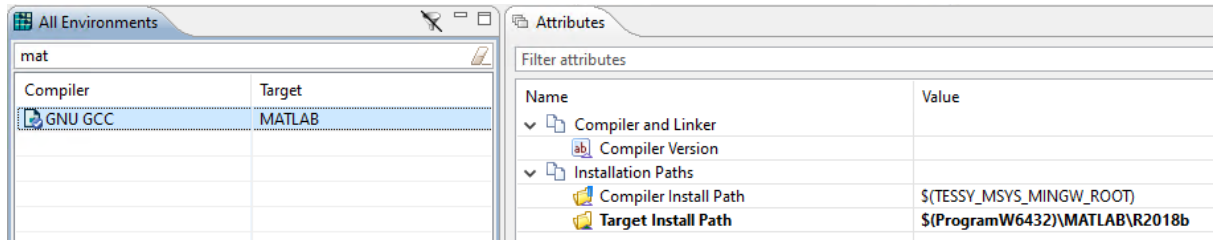
A prerequisite for testing is the C code generated from the model (either MATLAB Coder or Targetlink code generator). The C code file is fed to TESSY the same way as you would do for normal unit tests. Please consult the TESSY user manual for further notes about unit tests.

TESSY will then analyze the C code in order to build the interface database. Based on the C code interface, test data can be entered that will be used for testing of the MATLAB model. Usually all MATLAB model ports are mapped to global or external variables of the same name within the generated C code. So these are the inputs and outputs that you will see in TESSY.

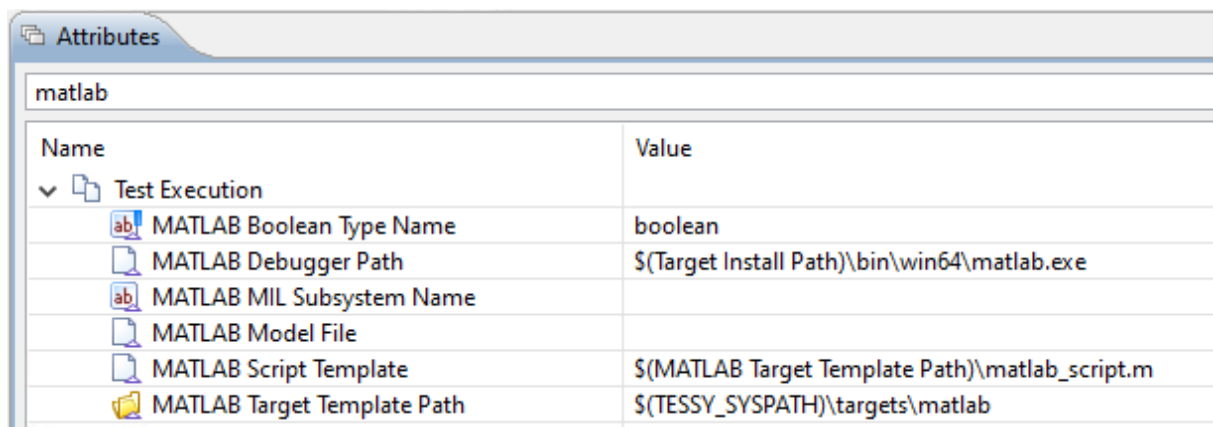
Please note: You should not run both MATLAB/Simulink and MATLAB/TargetLink test runs within the same batch test.

2 TESSY Environment Settings

Within the TEE perspective set the **Target Install Path** attribute to the installation directory of MATLAB:

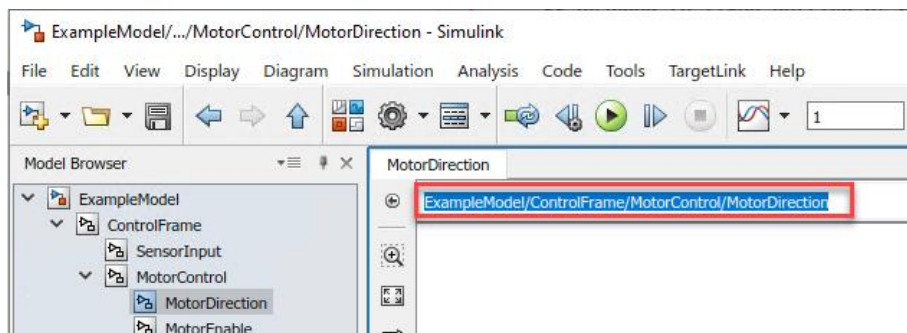


Within the “GNU GCC MATLAB” environment there are several attributes you have to pay attention to. These attribute names all start with MATLAB, e.g. **MATLAB Model File**, **MATLAB MIL Subsystem Name**. Only the latter ones are visible in non-expert mode of the TEE while in expert mode the following attributes are visible.



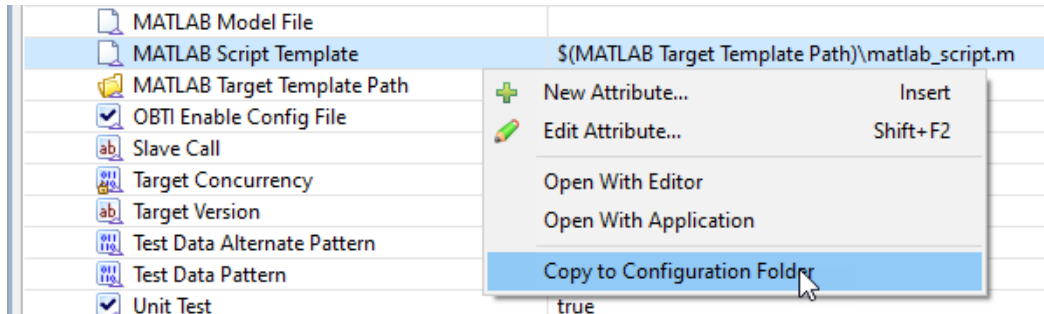
You can set the path to your MATLAB model file and set the subsystem name if there is any. Normally it is useful to set the MATLAB model file as well as the MIL subsystem path on module level within the **Attributes** tab of TESSY’s **Properties** view.

The **MATLAB MIL Subsystem Name** is the complete path to the MATLAB subsystem that shall be tested as shown within the MATLAB explorer bar:

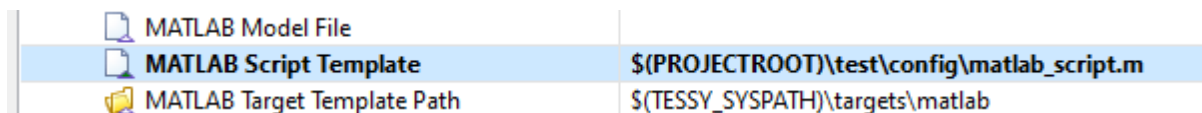


MATLAB Model Testing

If the MATLAB model needs any additional paths or other MATLAB settings to run, you can copy and modify the script template file using the **Copy to Configuration Folder** menu entry:



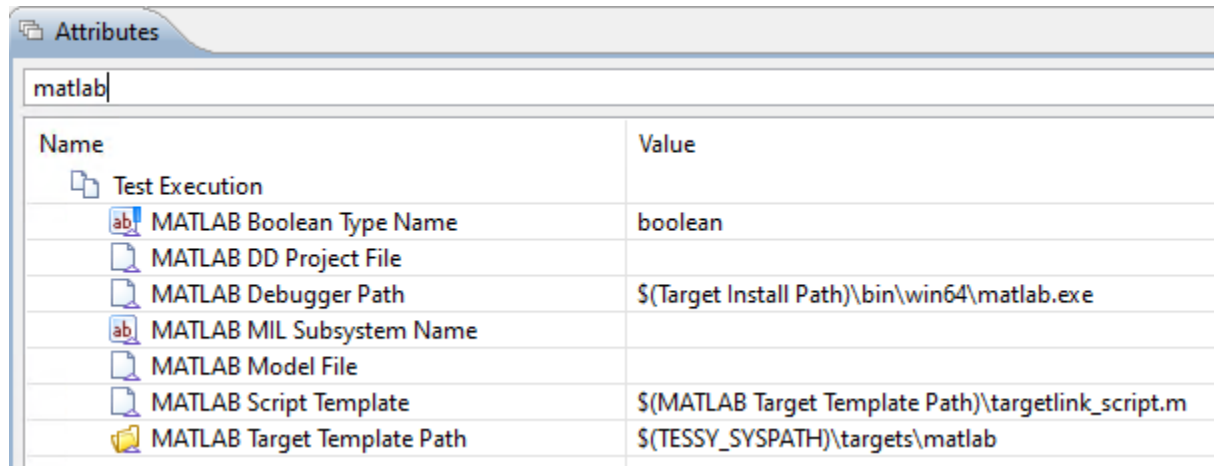
This will copy the file into your configuration folder where you can edit it safely.



For example you can use the command `addpath(genpath('C:/...'))`; to add a directory and all its subdirectories to the project. You can also run your own MATLAB scripts there. Refer to the MATLAB documentation for details.

3 dSPACE TargetLink

To run your TargetLink model, you need to choose the “GNU GCC TargetLink” environment. The environment settings are mostly the same, there is only one additional attribute:

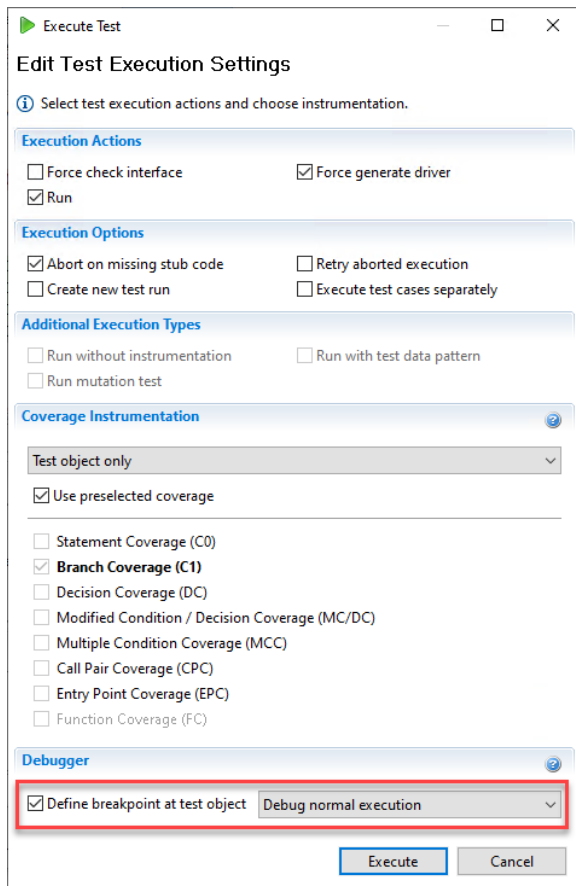


Name	Value
Test Execution	
MATLAB Boolean Type Name	boolean
MATLAB DD Project File	
MATLAB Debugger Path	\$(Target Install Path)\bin\win64\matlab.exe
MATLAB MIL Subsystem Name	
MATLAB Model File	
MATLAB Script Template	\$(MATLAB Target Template Path)\targetlink_script.m
MATLAB Target Template Path	\$(TESSY_SYSPATH)\targets\matlab

As with a MATLAB model it is useful to set the MATLAB model file, the MIL subsystem path as well as the **MATLAB DD Project File** on module level within the **Attributes** tab of TESSY’s **Properties** view.

4 Interactive Debugging

For interactive debugging select **Define breakpoint at test object** and click **Execute**.



TESSY will start the MATLAB GUI and open the test driver model. You can check the setup of the ports, add displays, add delays, or log signals, as with a normal Simulink model. When you have finished inspecting the test data, start the simulation in Simulink by clicking the **Run** button or enter the command `sim(miltest)` within MATLAB's **Command Window**.

5 Troubleshooting

If the MATLAB model test does not start or fails please watch out for error messages in TESSY's **Console** view.

It may help to increase the **Batchtest Wait Time** if one test run succeeds and another one fails. The default value is set to 2 seconds.