

MES Test Manager® (MTest)

The Model Test Manager

MES Test Manager® (MTest) is a test management tool that facilitates ISO 26262-compliant, requirements-based testing of Simulink®, Embedded Coder®, and TargetLink® models. MTest automates all test activities for unit and integration testing in terms of functional testing and regression testing. Furthermore, MTest supports all simulation types from model-in-the-loop to processor-in-the-loop simulation so as to support back-to-back testing as well.

MTest guarantees software quality assurance and compliance with standards such as the automotive industry's ISO 26262. In addition, it supports the ISTQB® methodology and techniques. MTest is the first choice for requirements-based testing as it simplifies development, improves quality, and ensures software safety. In the latest version of MTest, a new feature supports functional test case generation based on MARS requirements and allows users to automatically generate test cases to trigger model behavior.

Your MES Test Manager® benefits:

Efficiency

- Simple and intuitive test case specification with MTCD
- Automated test evaluation

Safety

- Assessment Framework for test result evaluation
- Complete documentation of tests and links to requirements

Control

- Highest degree of test automation
- Supports TargetLink® as well as Embedded Coder®

Compliance

- Traceability and coverage (model, code, requirements, etc.)
- Requirements-based testing in compliance with ISO 26262 and ISTQB

MTest Tool FAQ

General Questions

How do I contact the MTest support team?

What system requirements do I need to run MTest?

Which MATLAB® versions are MTest compatible with?

Which TargetLink® version does MTest support?

Where do I download the latest version of MTest?

How do I install MTest?

How do I configure my license?

Is MTest certified in any form?

Does MTest support test processes with respect to ISO 26262?

Specific Questions

Which types of models can be tested with MTest?

Which model parts can be tested with MTest?

How do I specify my test cases?

How does MTest test my model?

What does MTCD mean?

What does MARS mean?

Is it possible to generate test cases with MTest?

How do I generate test cases?

How should I structure my test suites?

How do I connect MTest with my Application Lifecycle Management (ALM) software?

How do I import my requirements into MTest?

How do I formalize requirements?

How do I link test cases to requirements?

How do I include measurement data for use within my test cases?

How do I record signals and states that are neither input nor output signals?

How do I generate and simulate my code?

How do I check if my model meets its requirements?

How do I check my requirements coverage?

How do I run my test projects automatically?

How do I export my test results?

Can I execute MTest and MXAM at the same time?

How do I track my project progress and integrity?

How MTest supports you:

1. MATLAB Integration

Matlab Integration

Image not found or type unknown

MTest is capable of supporting multiple projects. It is fully integrated in MATLAB[®] and supports releases from R2013b to R2021b. This allows MTest to access all MATLAB[®] variables and parameters relevant to model testing. MTest specializes in software testing of implementation models with dSPACE TargetLink[®], as well as Embedded Coder[®] of MathWorks.[®]

2. Easy import of requirements

Easy import of requirements

Image not found or type unknown

MTest imports requirements directly and easily, whether from a requirements workbook or a file. All requirements are checked for their testability. If requirements alter, MTest will signify the changes.

3. Efficient specification of test cases

Efficient specification of test cases

Image not found or type unknown

MES developed a clear and easily comprehensible test case specification language for MTest, Model Test Case Design Language (MTCD). MTCD allows efficient specification of all required signals and test cases to be individually linked to requirements.

If the test case specification requires other methods, other measurement data (ASCII, MAT, Excel), or classification trees by TESTONA/CTE (Berner & Mattner) can also be imported.

4. Automated test execution

Automated test execution

Image not found or type unknown

Automated test execution is a fundamental component of MTest. It includes test bed generation, model simulation according to input data, and recording results for MIL, SIL, and PIL simulation along with all internal (local) measurement parameters. All fully automated, totally hands-off! It is up to you whether you wish to set the pace or let MTest do all the tests for you with the convenient batch mode – either way, you are always in control.

5. Automated evaluation of test results

Automated evaluation of test results

Image not found or type unknown

The Assessment Framework is a powerful MTest tool to evaluate test results safely and automatically. This increases test efficiency and improves error detection.

The approach to check each requirement on all test sequences provides higher test coverage and consequently a reliable assertion regarding the functional quality of your software model. In this way, the Assessment Framework provides cost-effective yet efficient software quality assurance.

6. Test coverage and traceability

Test coverage and traceability

Image not found or type unknown

MTest automatically aggregates all relevant coverage data for requirements, test specifications, code, and models.

The Requirements, Coverage, Traceability (RCT) Framework integrated in MTest links coverage data directly with requirements. In this way, the traceability of results is continually possible.

7. Test documentation

Test documentation

Image not found or type unknown

With MTest, the documentation of test results is a top priority. MTest documents everything a model tester could wish for such as detailed test reports with signal graphics, applied parameters, and test results and compact test catalogs for a quick overview of what is most essential.

MTest even measures test progress and test quality. This helps quality managers to clearly see where action is required.