Reactis®
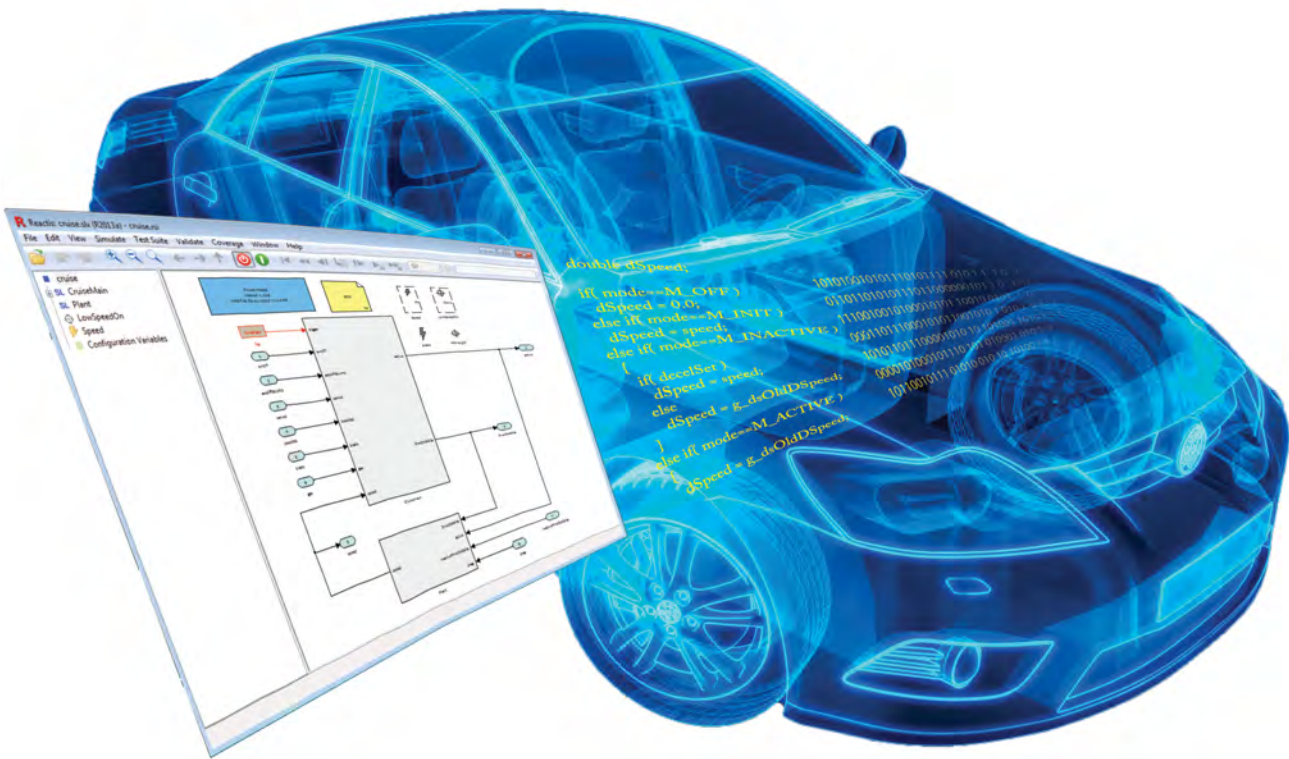# Product Guide

- Generate tests from Simulink® models or C code

- Detect runtime errors, check functional requirements

- Track coverage: decision, condition, MC/DC, and more

- Back-to-back testing of code against model

What is Reactis?

The Reactis product line supports testing and validation in a model-based design process that uses Simulink®, Stateflow®, Embedded MATLAB®, and C code. The line consists of five products:
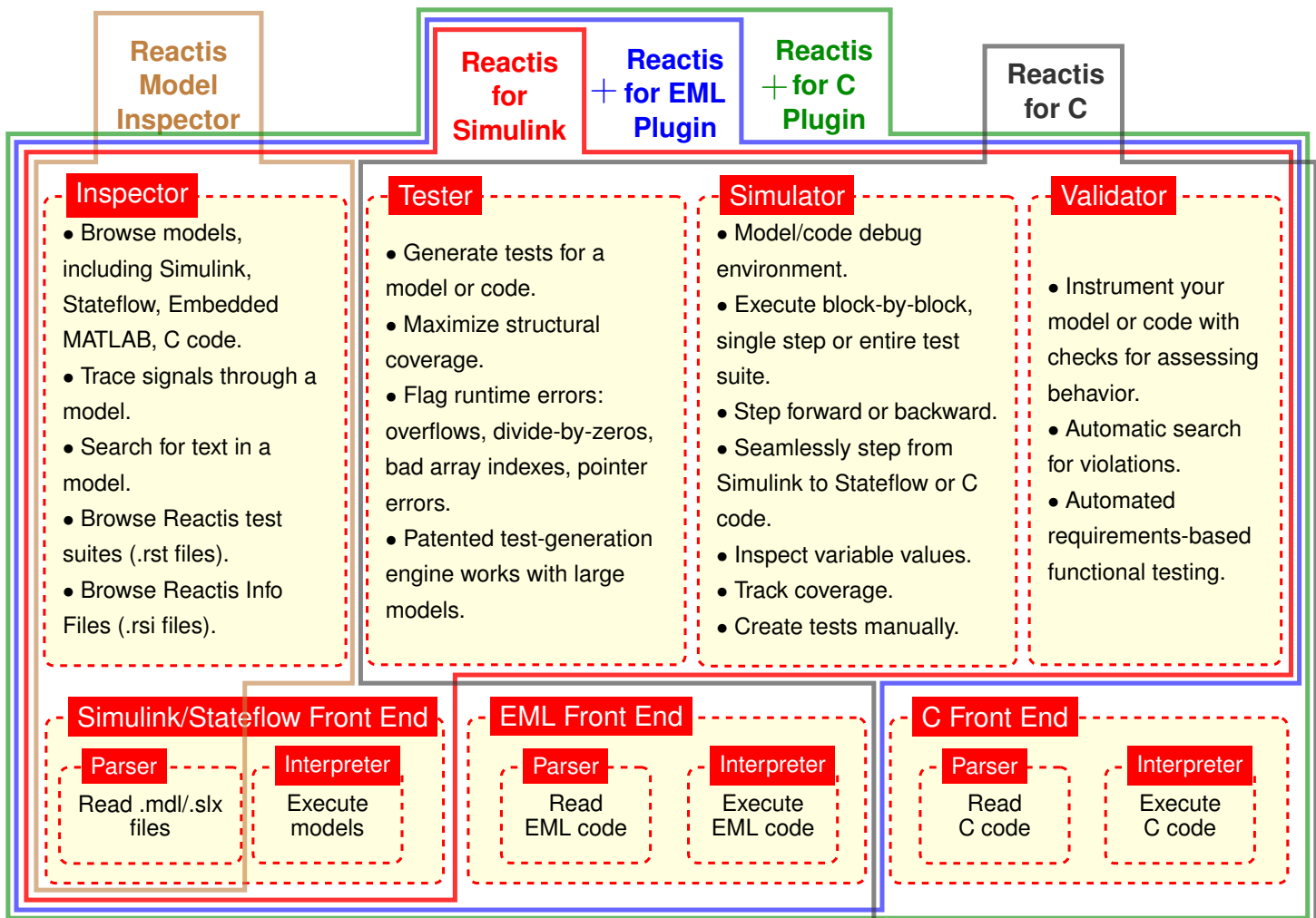
Reactis for Simulink Automates the generation of test data from, and validation of, Simulink/Stateflow models.

Reactis for C Plugin Integrates seamlessly with Reactis for Simulink to offer white-box analysis of the C code portions of models (S-Functions and Stateflow custom code).

Reactis for EML Plugin Integrates seamlessly with Reactis for Simulink to offer white-box analysis of the Embedded MATLAB (EML) code portions of models.

Reactis Model Inspector Is a low-cost, lightweight viewer for Simulink models and testing artifacts.

Reactis for C Offers advanced testing and validation capabilities for C code that is not incorporated in a model.

Reactis Model Inspector	Reactis for Simulink	+ Reactis for EML Plugin	+ Reactis for C Plugin	Reactis for C

Inspector
- Browse models, including Simulink, Stateflow, Embedded MATLAB, C code.
- Trace signals through a model.
- Search for text in a model.
- Browse Reactis test suites (.rst files).
- Browse Reactis Info Files (.rsi files).

Tester
- Generate tests for a model or code.
- Maximize structural coverage.
- Flag runtime errors: overflows, divide-by-zeros, bad array indexes, pointer errors.
- Patented test-generation engine works with large models.

Simulator
- Model/code debug environment.
- Execute block-by-block, single step or entire test suite.
- Step forward or backward.
- Seamlessly step from Simulink to Stateflow or C code.
- Inspect variable values.
- Track coverage.
- Create tests manually.

Validator
- Instrument your model or code with checks for assessing behavior.
- Automatic search for violations.
- Automated requirements-based functional testing.

Simulink/Stateflow Front End

Parser
Read .mdl/.slx files

Interpreter
Execute models

EML Front End

Parser
Read EML code

Interpreter
Execute EML code

C Front End

Parser
Read C code

Interpreter
Execute C code

Coverage-Based Testing

Reactis Tester employs a patented technology called *guided simulation* to build test suites. The tool tracks several coverage metrics as it computes test data, and it uses uncovered elements of these metrics to influence the creation of subsequent tests.

The coverage metrics include generic objectives such as decision, condition, and MC/DC, Simulink-specific measures such as branch and subsystem, and Stateflow-specific metrics such as state, condition action, and transition action.

As Reactis Tester generates tests, it performs an array of checks to flag runtime errors in your model or code. Common errors include overflows, divide-by-zeros, bad array indexes, and pointer errors.

How much of my model/code have I tested?

Will unexpected inputs cause a runtime error?

| Simulink model or C code | → | **Reactis Tester** | → | Test suite |

- *Maximize coverage*
- *Avoid redundancy*

Reactis Coverage Metrics

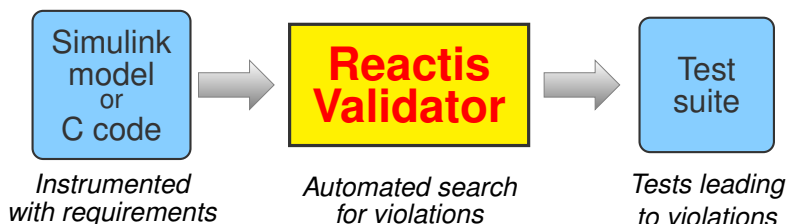| **Simulink**	**Stateflow**	**Generic**
• Branch	• State	• Statement
• Subsystem	• Condition action	• Decision
• Lookup Table	• Transition action	• Condition
	• Child state exit via parent transition (CSEPT)	• MC/DC
		• MCC

Functional Testing

Validator, the model-validation component of Reactis, uses a "test and check" approach for verifying models. You first instrument your model with checks that monitor whether model behavior is consistent with expectations. Validator then uses the test-generation component of Tester to create and run test data to see if any checks can be violated. These tests can then be used to diagnose the cause of any abnormal behavior detected.

Checks can be as simple as monitoring range constraints on individual signals or as complicated as determining whether or not a model satisfies a system-level requirement.

*Will the brake always deactivate the cruise control?*

| Simulink model or C code | → | **Reactis Validator** | → | Test suite |

Instrumented with requirements — *Automated search for violations* — *Tests leading to violations*

Back-to-Back Testing of Code Against Model

Testing C code (auto-generated or hand-written) against your model is easy with Reactis:

1. Generate tests from model
2. Run tests on code
3. Differences are automatically detected, flagged, and recorded in reports.

OR

4. Generate tests from code
5. Run tests on model
6. Differences are automatically detected, flagged, and recorded in reports.
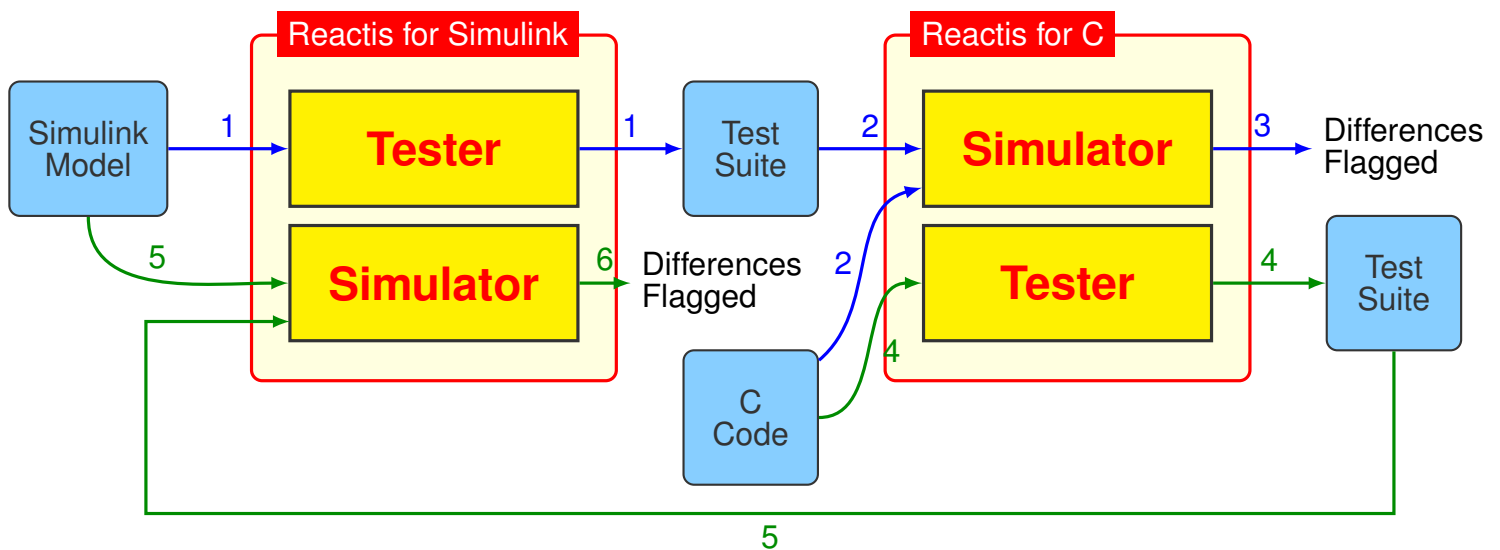
	Reactis Model Inspector	Reactis for Simulink	Reactis for EML Plugin	Reactis for C Plugin	Reactis for C
Software Dependencies	None	Requires MATLAB, Simulink, and Stateflow from MathWorks. Supported versions: R14-R2016a. Supports TargetLink versions: 2.1.6-4.0.	Requires Reactis for Simulink	Requires Reactis for Simulink	None
Platforms	Windows XP, Vista, 7				