

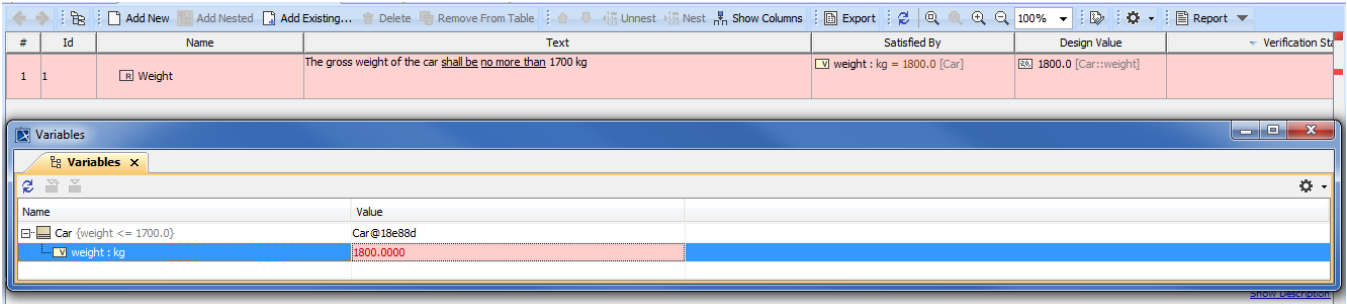
Recording Verification status of runtime values

On this page

- Remember Failure Status option

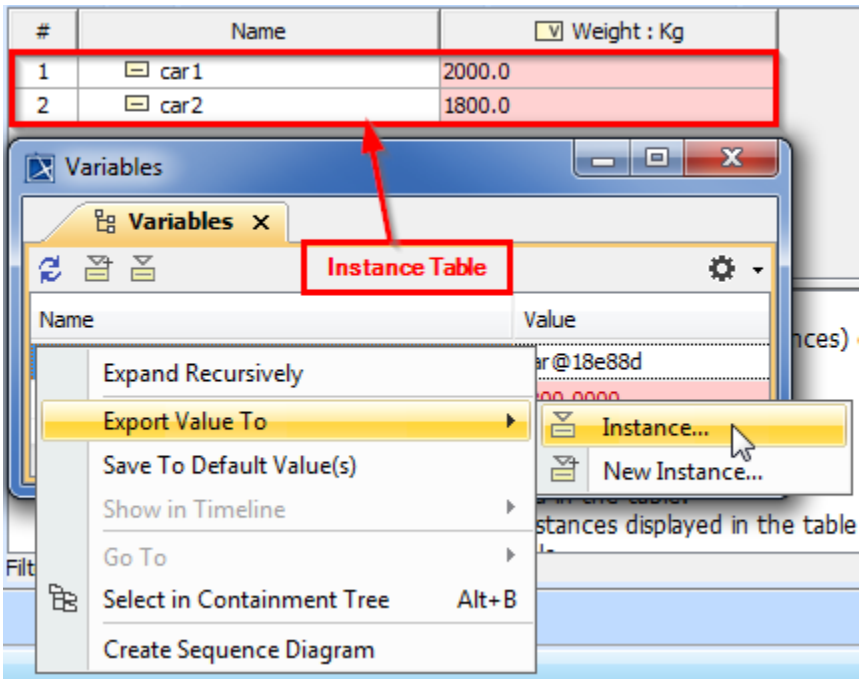
Magic Model Analyst supports recording verification status of runtime values. You can verify and record the status of constraints during runtime. If constraints are not met during the simulation, cells in the Requirement/Instance table will be highlighted in red on the **Variables** pane. You can do the normal recording verification status of runtime values by using the [resultLocation](#) tag in [SimulationConfig](#). The instances with recorded values in slots are created at the end of simulation.

The Requirement(s) table shows that the constraint in the requirement is not met, thus the requirement is colored in red as follows



The Requirement(s) table showing the constraint within a requirement which is not met and colored in red.

If you right-click on the **Variables** pane and select **Export Value To > Instance**, the variables will be evaluated and colored in red if there are constraint failures in the instance table, as shown in the following figure



The instance table showing the constraints after being exported to an Instance.

The status (pass/fail), related requirements, and instances for constraints are recorded using the **VerificationStatus** stereotype that is applied on the **Value** specification of the properties that you are working on. You can also see the detail of the constraint failure in the tooltip when hovering the mouse over any highlighted red values in the Requirement/Instance table as shown in the following figure

Variables x Breakpoints x	
Name	Value
Car {weight <= 1700.0, high <= 5.0}	Car@66714a
high : meter	3.0000
weight : kg	1800.0000
highCheck : High Verification {high <= 5}	HighVerification@1daca93
Requirement 1 - "The gross weight of the car shall be no more than 1700 kg" is not satisfied.	
weightCheck : Weight Verification {weig...	WeightVerification@558ec0

The simulation console showing the variables which have met or not met the constraints.

In addition, failed verdict of **State Invariant** is highlighted with red values and the tooltip. However, you must create a Dependency relation between the testcase and value property as shown in the following figure.

WithClock results x

Criteria
Classifier: Test Scope (optional): :tResult: {key} Filter:

#	Name	system : System	tc1 : VerdictKind
1	test at 2018.01.10 12.15	test.system8 : System	fail
2	test at 2018.01.10 12.16	test.system9 : System	fail
3	test at 2018.01.10 12.20	test.system : System	fail
4	test at 2018.10.05 10.42	test.system10 : System	fail
5	test at 2018.10.05 10.43	test.system11 : System	fail
6	test at 2018.10.05 10.52	test.system13 : System	fail

Model x

bdd [Package] SimpleTestWithClock [Model]

«block» Test

values

tc1 : VerdictKind = inconclusive

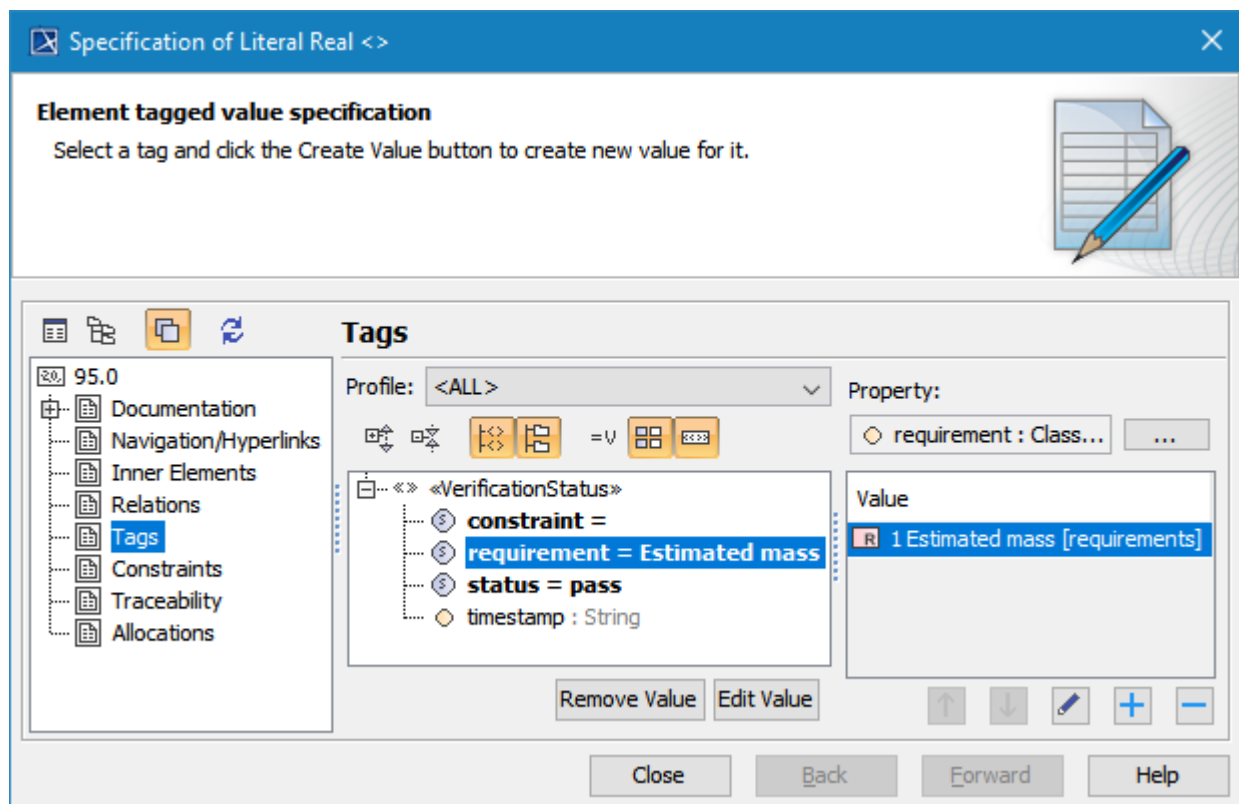
d1

«block» System

«testCase» testcase1

Failed State Invariant with highlighted red values and the tooltip.

If the **VerificationStatus** state on the **Value** specification fails, the active validation will be marked for notification with the **fail** status tag as shown in the following figure



The Specification dialog showing verification status with the fail status tag.

The **VerificationStatus** stereotype, as shown in the figure above, has the tags as follows

Tag	Description
constraint	The name of the failure constraint.
requirement	The name of the failure Requirement.
status	The result of the verification, either pass or fail.
timestamp	The timestamp of the first failure recorded.

Remember Failure Status option

During the evaluation of particular configurations and scenarios, some Requirements/constraints can fail, and those values are marked in red. However, they may pass later and are marked in green again, but the failure is not recorded by default. To solve this problem, Simulation Toolkit provides the **Remember Failure Status** option in «SimulationConfig» to remember any failure status until the termination and record it as fail, even though it passes at the end. If the **Remember Failure Status** option is set true, and there are constraint failures during the simulation, the first fail status will be recorded in the status of «VerificationStatus» of new/recorded instance, and the time of the first failure will also be recorded in the **timestamp** tag as shown in the figure below.

«SimulationConfig»
Tank Control
 «SimulationConfig»
 UI = TimeSeries
 addControlPanel = false
 animationSpeed = 95
 autoStart = true
 autostartActiveObjects = true
 cloneReferences = false
 constraintFailureAsBreakpoint = false
 endTime = 15000
 executionTarget = system
 fireValueChangeEvent = true
 initializeReferences = false
 numberOfRuns = 1
 rememberFailureStatus = true
 resultLocation = results
 runForksInParallel = true
 silent = true
 solveAfterInitialization = true
 startTime = 0
 startWebServer = false
 stepDelay = 1.0
 stepSize = 20.0
 timeUnit = millisecond
 timeVariableName = "simtime"
 treatAllClassifiersAsActive = true

System

waterTank.level, waterTank.valve

3
2.75

waterTank.level fails

waterTank.level fails

waterTank.level fails

Specification of Literal Real <>

Element tagged value specification
 Select a tag and click the Create Value button to create new value for it.

Tags

Profile: <ALL>

Property: status : EnumerationLiteral

Value
 fail [SysML::Requirements::VerdictKind]

fail

Remove Value Edit Value

Close Back Forward Help

control : Controller system.waterTank.Control : ...
 waterTank : WaterTank system.waterTank : WaterT...
 : level {waterTank.level <= co... level@4b0d5858
 level : Real 1.1346
 maxLevel : Real 3.0000

00:00:00,000 : **** Instance Specification system is started! ****
 00:00:02,980 : Constraint level {waterTank.level <= control.maxlevel} failed.
 00:00:07,960 : Constraint level {waterTank.level <= control.maxlevel} failed.
 00:00:12,920 : Constraint level {waterTank.level <= control.maxlevel} failed.

>> (default)

Setting the Remember Failure Status tag of a SimulationConfig to record the fail status and timestamp if available.

Related pages

- Requirement Table
- Variables pane