

SimulationConfig stereotype

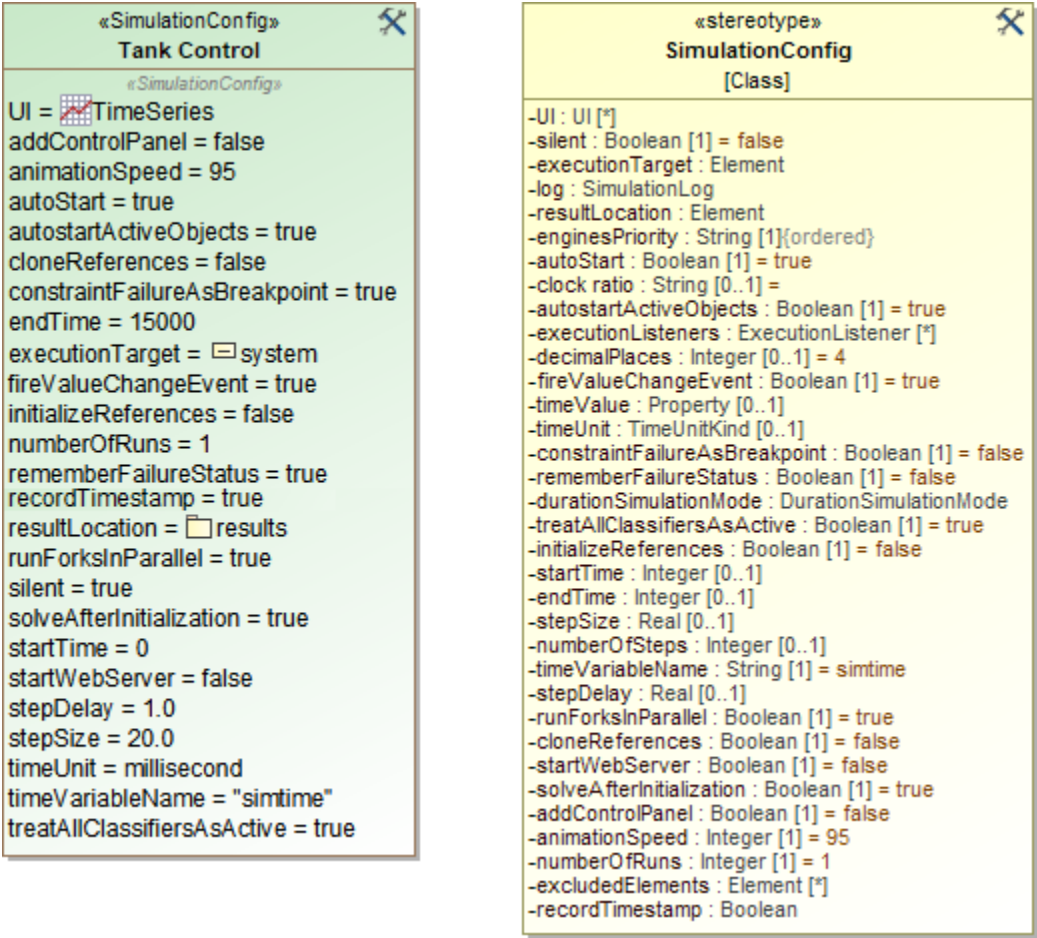
On this page

- [Initialize References property](#)
- [Instantiation scope with excluded elements](#)

Magic Model Analyst provides a model-based execution configuration through the «stereotype» *SimulationConfig*. You can configure the stereotypes in the Specification window of «SimulationConfig». The following figure shows the «SimulationConfig» and the SimulationConfig stereotypes

To show up the «SimulationConfig» and the SimulationConfig stereotype




1. Right-click a Package in the Containment tree in a simulation project and select **Create Diagram > Simulation Configuration Diagram**, which is the SimulationProfile stereotype applied.
2. Right-click the *SimulationProfile* [SimulationProfile.mdzip] stereotype in the Containment tree and select **Config > SimulationConfig [Class]** and drag the Class to the diagram pane.




The «SimulationConfig» and its «stereotype».

The *SimulationConfig* «stereotype» contains tag definitions. You can sort these tag definitions alphabetically by right-clicking it, and select **Symbol Properties > Attributes Sort Mode > By Name**. Click the «SimulationConfig» to open its Specification window and see all of the tag definitions.

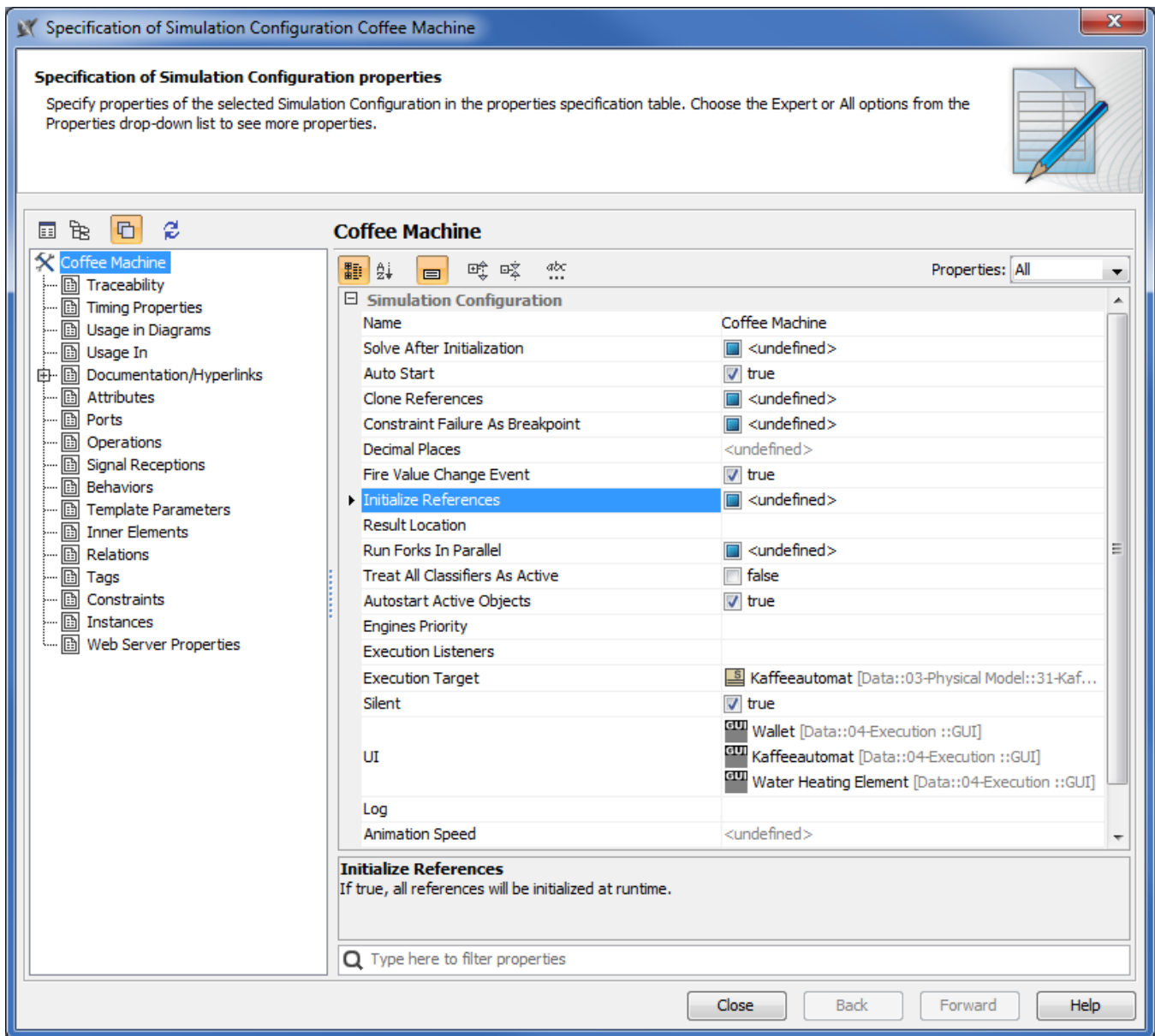
Tag definition	Description
UI	A user interface for configuration mockups that will start with a model simulation.
silent	If the value is true , simulation will run without animation or idle time.

executionTarget	<p>An element from which simulation should start.</p> <div>  Note You can drag any element that can be simulated to a Simulation Configuration to set executionTarget for the dragged element. </div>
excludedElement	A list of elements which will be excluded and not instantiated if not ready to be used or not needed, e.g., Class, Package, Use Case, Actor, Behavior, Connector, Port, and State. See also in Instantiation scope with excluded elements .
log	An element in which the execution trace will be recorded.
resultLocation	<p>A Package, a Model, InstanceSpecification, or an InstanceTable in which a context object will be stored after simulation.</p> <ul style="list-style-type: none"> If the resultLocation is specified by an InstanceSpecification, the values of the context object will be saved as slot values of the specified InstanceSpecification. If the resultLocation is specified by a Package or a Model, Magic Model Analyst will create a new InstanceSpecification owned by the Package or the Model after the simulation. The values of the context object will then be saved as slot values of the created InstanceSpecification. If the resultLocation is specified by an InstanceTable, a new result instance will be created and added as a new row with only visible columns of the table. However, if the executionTarget and the resultLocation are the same InstanceTable, the result instance will be updated instead (only visible columns of the table). See also Instance table in MagicDraw User Guide. <div>  Note The following elements will not be saved </div> <ul style="list-style-type: none"> InstanceTable without the Classifier set. If the resultLocation is not specified, the simulation results will not be saved even though the simulationTarget is the InstanceSpecification. Nested instances inside slots.
enginesPriority	Simulation engines that can be used to simulate a model ordered by priority. The first engine on the list has the highest priority. If the Simulation Configuration does not have a tagged value for this tag definition, Magic Model Analyst will use the values that are defined in the registered Simulation Engine Priority in the Project Options dialog.
autoStart	<p>If true, a model simulation starts running automatically once it has been initialized. Otherwise, you must click the Start button in the Simulation pane to run the simulation.</p> <div>  Note For Monte Carlo Simulation and Table Simulation, the conditions are as follows: </div> <ul style="list-style-type: none"> If true, the simulation will be initialized, solve parametric, start Behaviors, wait for Behaviors terminated or when endTime is reached, and terminate those Behaviors are Blocks with Classifier Behaviors and/or Part Property with Behaviors. If endTime is less than the time of part Behaviors, e.g., duration constraints are applied, the time of each iteration will be endTime. However, when endTime is more than the time of one Behaviors, all part Behaviors will be completed just once. If false, the simulation will be initialized, solve parametric, and terminate. Behaviors will not be started.
clock ratio	A ratio between a simulation clock and a real-time clock (1:10). For example, if the clock ratio is 10, it means that one second on the simulation clock is equal to 10 seconds on the real-time clock.
autostartActiveObjects	<p>If the value is true, the runtime objects whose classifier is active will start their behavior automatically in an asynchronous mode. Otherwise, the behavior will start using a startObjectBehaviorAction.</p> <ul style="list-style-type: none"> If false, the simulation will be initialized, solve parametric, and terminate. Behaviors will not be started.
executionListeners	A list of execution listeners that will receive events from a model simulation. An execution listener can be a SequenceDiagramGeneratorConfig .
decimalPlaces	Decimal places of all displayed numerical values in a model simulation, for example, in the Variables pane, Tooltip, and the SimulationConsole pane. Their values must be integers. If the precision of displayed numerical values is greater than or equal to 10% of the absolute value, the numerical values will be displayed in exponential form.
fireValueChangeEvent	If this value is true , a parametric simulation will be repeated immediately whenever the value of any structural feature changes. Repeating simulation will give an impact on both values carried over by binding connectors and specified in a constraint.
timeValue	A property that will be used in a model-based clock. If the tagged value is specified, the run-time value specified by the property will be used as the simulation time.
timeUnit	A unit of a runtime value that specifies the simulation time. If the tagged value is unspecified, the millisecond will be used by default.
constraintFailureAsBreakpoint	A flag that indicates a simulation will pause at the constraint element where the failure occurs.
rememberFailureStatus	If true , the failure status will be remembered until the termination, and the first failure status will be recorded with its timestamp as fail, even though it passes at the end.
durationSimulationMode	Specifies a duration mode of simulation in min, max, average, or random. Magic Model Analyst uses the duration mode to calculate the duration of an Action simulation to which a duration constraint is applied.

treatAllClassifiersAsActive	This option (true by default) runs all Behaviors, including the ones in the subsystems. Even though the isActive property is false , if TreatAllClassifiersAsActive is true, Magic Model Analyst will run all of the Behaviors of non-active Classifiers.
initializeReferences	If true , references will be initialized by creating new objects, instead of referencing to existing parts of the system. See also in Initialize References property .
startTime	Specifies when simulation will initiate (in integer).
endTime	The time simulation terminates (in integer).
stepSize	Increases as simulation time does (in real numbers or decimal places).
numberOfSteps	Specifies the number of steps through which simulation will run.
numberOfRuns	A number of runs of a particular configuration, especially for the sample size in Monte Carlo simulation. The value must be greater than or equal to 1.
timeVariableName	This property returns time from the simulation clock (the default value is "simtime"). It returns the time on the simulation clock.
stepDelay	Specifies how long the delay will be, if any, for each step.
runForksInParallel	If the value is true or undefined, all outgoing edges from the fork will run in parallel.
cloneReferences	If true , creates new instances for all objects recursively. Otherwise, it creates instances only for the first level and others remain in their original locations (the default is false).
startWebServer	If true , starts a web server for a remote web-based UI mockup. Note that the HTML UI must be generated first using the Generate HTML button in the UI diagram. After starting SimulationConfig, the web server URL will appear in the Simulation Console window.
solveAfterInitialization	If true , automatically starts initial solving. If false, manual start is required via Refresh in the Variables pane, or press Start (F8) .
addControlPanel	If the value is true , adds the simulation control panel into the exported HTML UI.
animationSpeed	Specifies the animation speed of a particular configuration in percentage. The default value is 95, but you can enter your preferred value from 1 to 100.
recordTimestamp	<p>If true, timestamp is recorded into a result instance name.</p> <div data-bbox="287 1167 1109 1207" data-label="Text"> <p> Note The naming of the newly created instance will not follow MagicDraw instance naming criteria.</p> </div> <p>If false (by default), MagicDraw instance naming will be used instead, and the timestamp will not be recorded and appended to the instance name.</p>

Initialize References property

The property Initialize References in the Specification window of «SimulationConfig» allows Magic Model Analyst to initialize all references between objects owned in a composite owner. The following is the Specification window of «SimulationConfig» from the [CoffeeMachine.mdzip](#) sample.



Initialize references in the Specification window of «SimulationConfig» Coffee Machine.



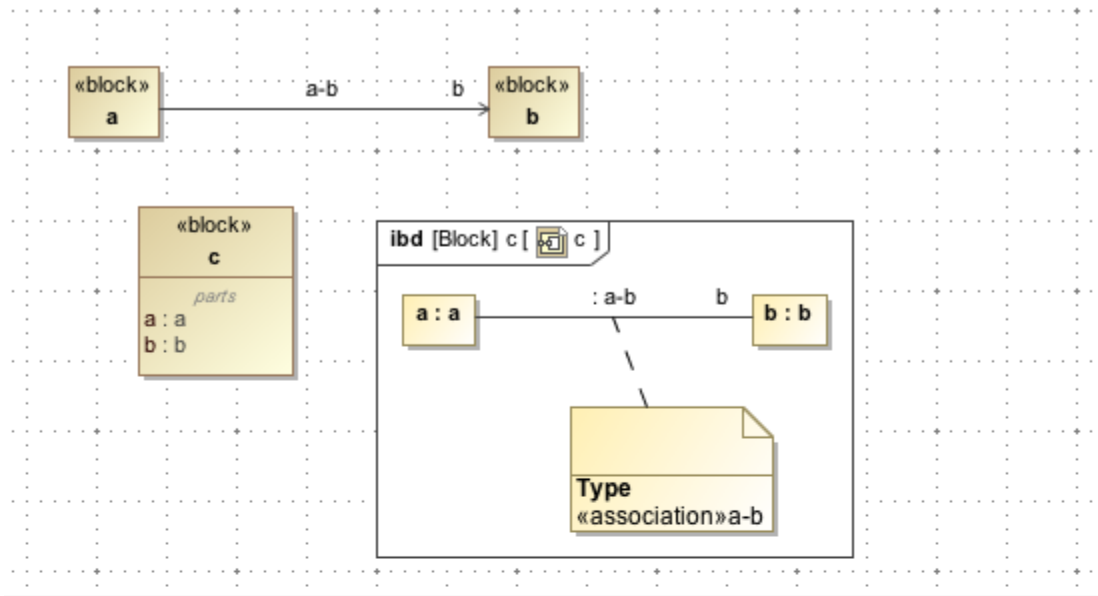
Note

Initialize References appears if you open the Specification window in the **All** or **Expert** Properties mode.

The following are two types of references that you can initialize using the Initialize References option in SimulationConfig

- **References between objects created inside a system**

This type of reference occurs when you create a project similar to the following example. The following figure shows «**block**» **c** as a root context, **a** is a Part Property, and **b** is both the part and reference properties. The «**block**» **c** contains an Internal Block diagram with a connector line (the type of the connector is association named **a-b**) that connects two part properties **a:a** to **b:b**. The connector specifies that reference property **b** will get the run time object value from Part Property **b**. If there is no connector, an auto-generated value will be given to reference property **b**.



Variables	
Name	Value
c	c@78bdbadb
a : a	a@74e720b1
b : b	b@5e546cf6
b : b	b@5e546cf6

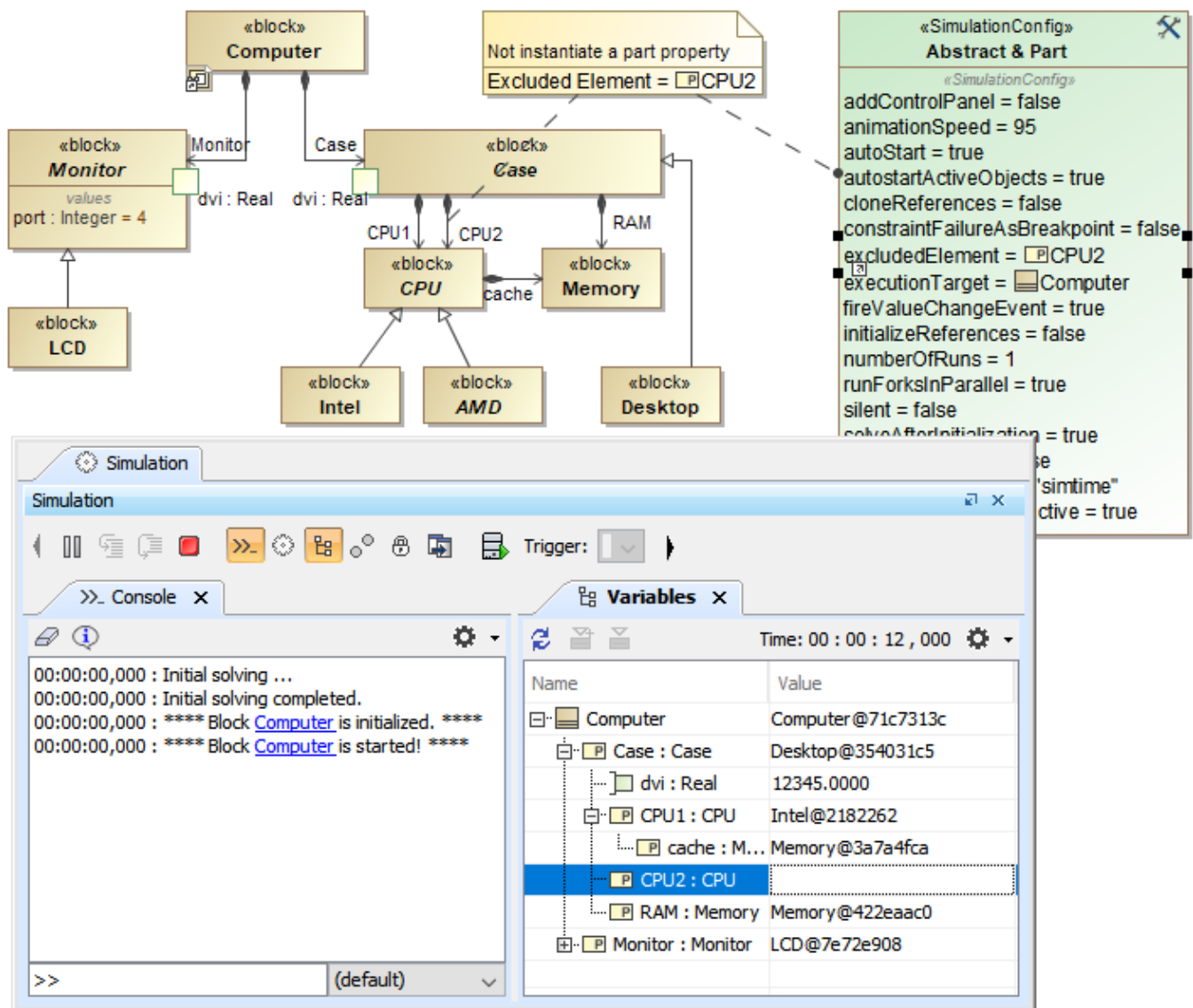
Initialization of Property.

- References to external objects

If you run a block, for example, the **Variables** pane will show the classifiers of the references. When the `InitializeReferences` is false, the reference property in the block will not be initialized. When it is true, the reference property will be initialized at run time.

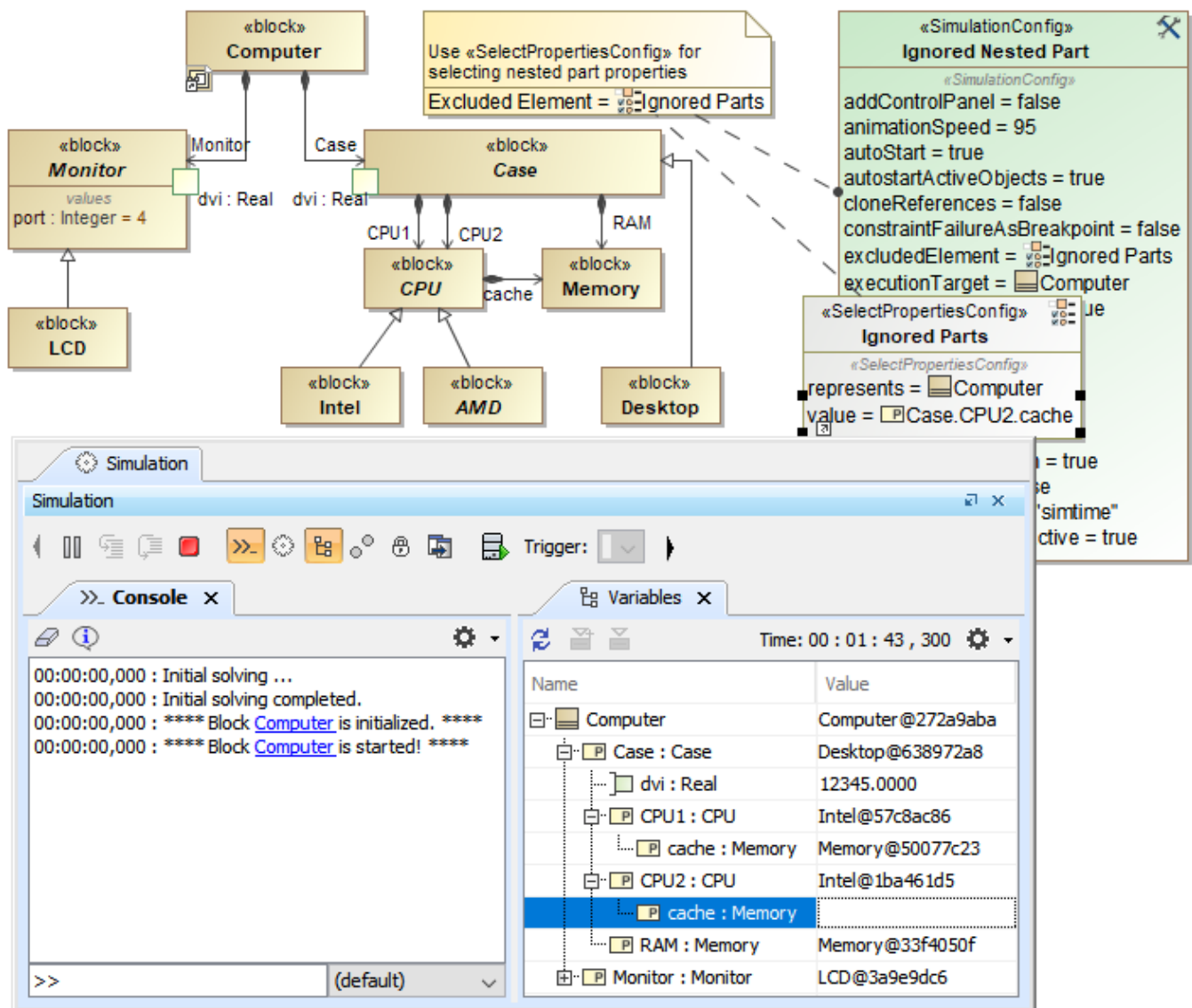
Instantiation scope with excluded elements

To control what part of the model is used in simulation, you can use **excludedElement** to exclude and not to instantiate any elements that are not ready to be used or not needed. When any Classifiers, e.g., Classes, Packages, Use Cases, Actors, Behaviors, Connectors, Ports, or States are selected, they will not be instantiated by default instantiation. Excluded Classifiers will not be displayed in the list of types in the **Variables** pane when clicked or right-clicked to select **Add value**. For example, a Part property is set in **excludedElement**, so it will not be instantiated when running a simulation as shown in the figure below. See also in [Automatic instantiation of the implementation Class](#).



Part property, CPU2, is not instantiated in the Variables pane, since it has been set in excludedElement.

For nested Part properties, **excludedElement** can be set through «SelectPropertiesConfig» as shown in the figure below. You can find the «**SelectPropertiesConfig**» control in the **Simulation Configuration Diagram** toolbar.



Nested Part property, cache of CPU2, is not instantiated in the Variables pane, since it has been set in excludedElement through «SelectPropertiesConfig».

Related pages

- [SimulationConfig stereotype](#)
- [Simulation log](#)
- [Simulation time and simulation clock](#)
- [Automatic start of active objects](#)
- [UI modeling diagram simulation](#)
- [ImageSwitcher and ActiveImage](#)
- [Time series chart](#)
- [Timeline chart](#)
- [Histogram](#)
- [Nested UI Configuration stereotype](#)
- [Reusable UI Mockup](#)
- [SimulationConfig stereotype](#)
- [CSV export](#)
- [Nested property selection for configurations](#)
- [Instance table](#)