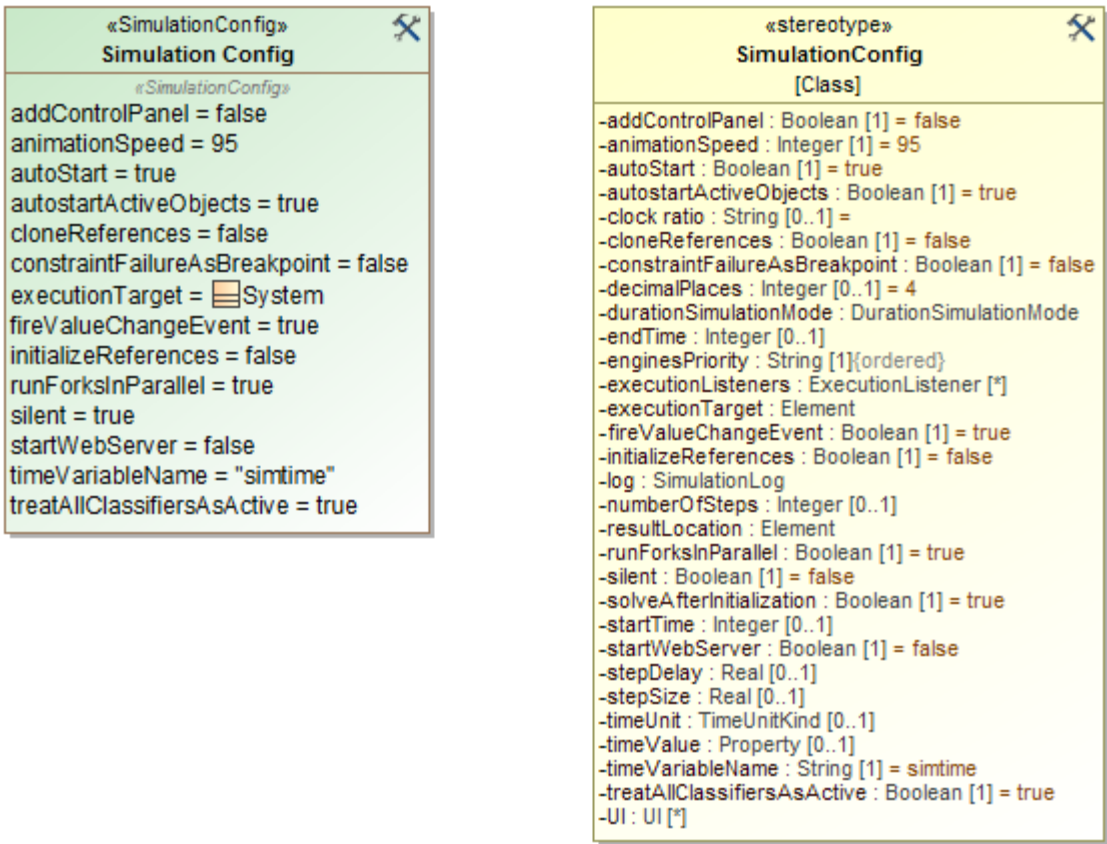


# SimulationConfig stereotype

Cameo Simulation Toolkit 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 Simulation Config stereotypes.

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	An element from which simulation should start.
log	An element in which the execution trace will be recorded.

<b>resultLocation</b>	<p>A <a href="#">Package</a>, a Model, <a href="#">InstanceSpecification</a>, or an <a href="#">InstanceTable</a> in which a context object will be stored after simulation.</p> <ul style="list-style-type: none"> <li>If the <b>resultLocation</b> is specified by an <a href="#">InstanceSpecification</a>, the values of the context object will be saved as slot values of the specified InstanceSpecification.</li> <li>If the <b>resultLocation</b> is specified by a <a href="#">Package</a> or a Model, Cameo Simulation Toolkit will create a new <a href="#">InstanceSpecification</a> 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.</li> <li>If the <b>resultLocation</b> is specified by an <a href="#">InstanceTable</a>, the values of the context object will be only saved as slot values of the predefined columns. Other existing slots, if any, remain the same (not deleted). Only the slots for primitive values represented as columns must be overwritten. See also <a href="#">Instance table</a> in MagicDraw User Guide.</li> </ul> <div style="border: 1px solid #f9e79f; padding: 10px; margin-top: 10px;"> <p> <b>Note</b> The following elements will not be saved</p> <ul style="list-style-type: none"> <li>InstanceTable without the Classifier set.</li> <li>If the <b>resultLocation</b> is not specified, the simulation results will not be saved even though the simulationTarget is the <a href="#">InstanceSpecification</a>.</li> <li>Nested instances inside slots.</li> </ul> </div>
<b>enginesPriority</b>	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, Cameo Simulation Toolkit will use the values that are defined in the registered Simulation Engine Priority in the <b>Environment Options</b> dialog.
<b>autoStart</b>	<p>If true, a model simulation starts running automatically once it has been initialized. Otherwise, you must click the <b>Start</b> button in the <b>Simulation</b> pane to run the simulation.</p> <div style="border: 1px solid #f9e79f; padding: 10px; margin-top: 10px;"> <p> <b>Note</b> For <a href="#">Monte Carlo Simulation</a> and <a href="#">Table Simulation</a>, the conditions are as follows:</p> <ul style="list-style-type: none"> <li>If true, the simulation will be initialized, solve parametric, start Behaviors, wait for Behaviors terminated or when <b>endTime</b> is reached, and terminate. Those Behaviors are Blocks with Classifier Behaviors and/or Part Property with Behaviors. If <b>endTime</b> is less than the time of part Behaviors, e.g., duration constraints are applied, the time of each iteration will be <b>endTime</b>. However, If the value is true, the runtime objects whose classifier is active will start their behavior automatically in an asynchronous mode. Otherwise, when <b>endTime</b> is reached, the time of each Behavior Action will be completed just once.</li> <li>If false, the simulation will be initialized, solve parametric, and terminate. Behaviors will not be started.</li> </ul> </div>
<b>clock ratio</b>	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.
<b>autostartActiveObjects</b>	<p>If the value is true, the runtime objects whose classifier is active will start their behavior automatically in an asynchronous mode. Otherwise, when <b>endTime</b> is reached, the time of each Behavior Action will be completed just once.</p> <ul style="list-style-type: none"> <li>If false, the simulation will be initialized, solve parametric, and terminate. Behaviors will not be started.</li> </ul>
<b>executionListeners</b>	A list of execution listeners that will receive events from a model simulation. An execution listener can be a SequenceDiagramGeneratorConfig.
<b>decimalPlaces</b>	Decimal places of all displayed numerical values in a model simulation, for example, in the <b>Variables</b> pane, Tooltip, and the <b>Simulation Console</b> 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.
<b>fireValueChangeEvent</b>	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.
<b>timeValue</b>	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.
<b>timeUnit</b>	A unit of a runtime value that specifies the simulation time. If the tagged value is unspecified, the millisecond will be used by default.
<b>constraintFailureAsBreakpoint</b>	A flag that indicates a simulation will pause at the constraint element where the failure occurs.
<b>durationSimulationMode</b>	Specifies a duration mode of simulation in min, max, average, or random. Cameo Simulation Toolkit uses the duration mode to calculate the duration of an Action simulation to which a duration constraint is applied.
<b>TreatAllClassifiersAsActive</b>	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, Cameo Simulation Toolkit will run all of the Behaviors of non-active Classifiers.
<b>InitializeReferences</b>	If true, references will be initialized by creating new objects, instead of referencing to existing parts of the system.
<b>startTime</b>	Specifies when simulation will initiate (in integer).
<b>endTime</b>	The time simulation terminates (in integer).
<b>stepSize</b>	Increases as simulation time does (in real numbers or decimal places).
<b>numberOfSteps</b>	Specifies the number of steps through which simulation will run.

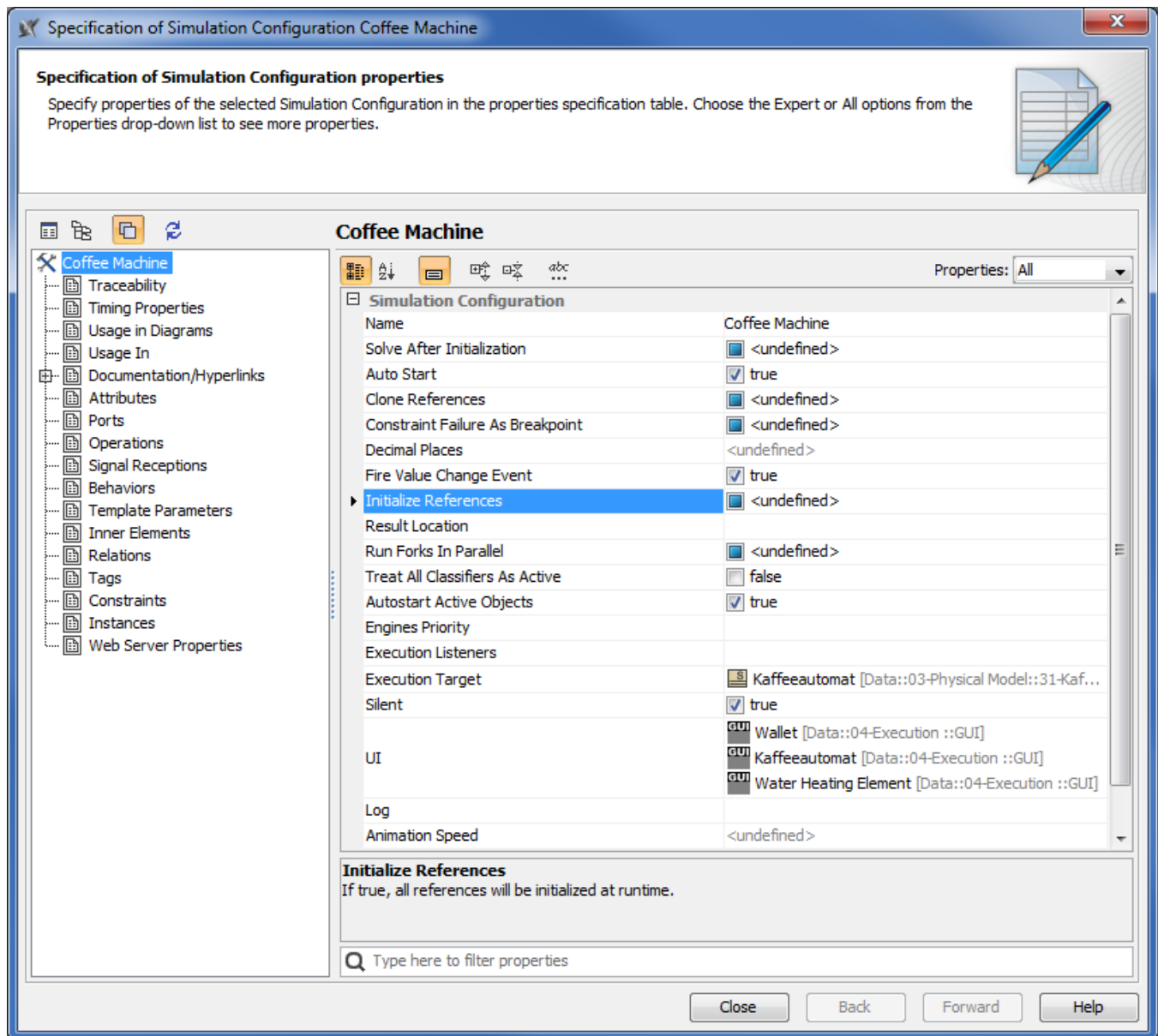
<b>numberOfRuns</b>	A number of runs of a particular configuration, especially for the sample size in Monte Carlo simulation. The value must be greater than or equal to 1.
<b>timeVariableName</b>	This property returns time from the simulation clock (the default value is "simtime"). It returns the time on the simulation clock.
<b>stepDelay</b>	Specifies how long the delay will be, if any, for each step.
<b>runForksInParallel</b>	If the value is true or undefined, all outgoing edges from the fork will run in parallel.
<b>cloneReferences</b>	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).
<b>startWebServer</b>	If true, starts a web server for a remote web-based UI mockup. Note that the HTML UI must be generated first using the <b>Generate HTML</b> button in the UI diagram. After starting SimulationConfig, the web server URL will appear in the <b>Simulation Console</b> window.
<b>solveAfterInitialization</b>	If true, automatically starts initial solving. If false, manual start is required via Refresh in the <b>Variables</b> pane, or press <b>Start (F8)</b> .
<b>addControlPanel</b>	If the value is true, adds the simulation control panel into the exported HTML UI.
<b>animationSpeed</b>	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.



#### Information

You can drag any element that can be simulated to a Simulation Configuration in order to set a simulationTarget (a simulationTarget tag will be set to the dragged element.).

The property Initialize References in the Specification window of «SimulationConfig» allows Cameo Simulation Toolkit 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.



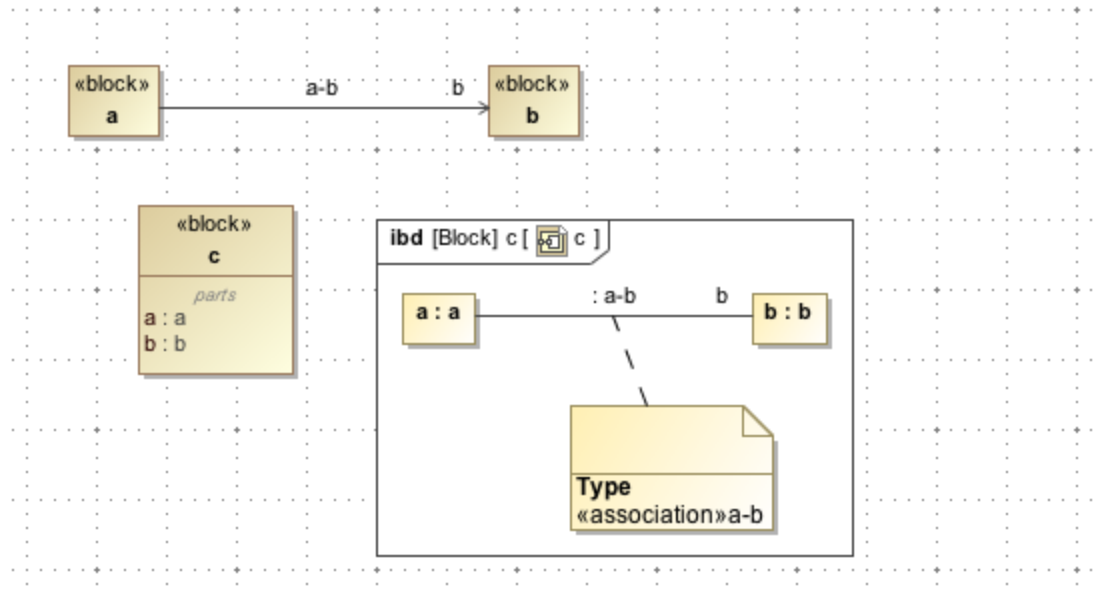
**Note**

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.

#### 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