

Recording simulation as a Sequence diagram

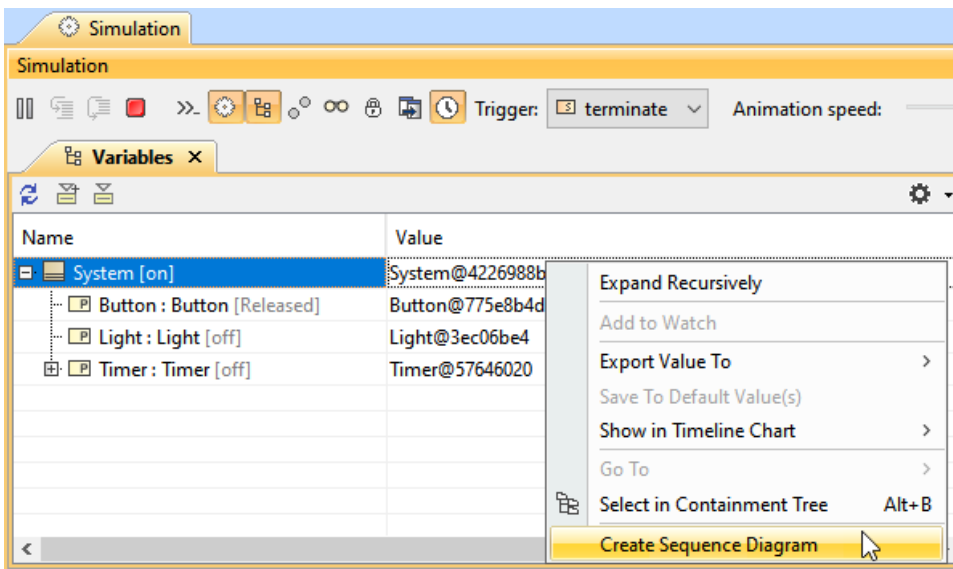
The recording capability of the Magic Model Analyst allows you to:

- record created objects as CreateMessages connected between Lifelines that represent the object creator and features of the created object respectively.
- record signals as SendMessages connected between Lifelines that represent signal senders and signal receivers respectively. Connectors will be assigned to the messages if signals are sent via ports or connectors.
- record operation calls as CallMessages connected between Lifelines that represent operation callers and operation owners respectively. Connectors will be assigned to messages if operations are called via ports.
- record changes of states and primitive values as StateInvariants on Lifelines that represent features of objects that own the states or the values.

This section demonstrates how to record the signal, state change, operation call, and value change as a sequence diagram during a model simulation. The sample `FlashingLight.mdzip`, located in the `<install_root>/samples/simulation/` directory, will be used throughout this section.

To record signals sent from and to a runtime object and subsequent state/value changes of the related objects as a sequence diagram

1. In the **Variables** pane, select and right-click a runtime object.
2. Click **Create Sequence Diagram** on the context menu. An empty sequence diagram will be created.



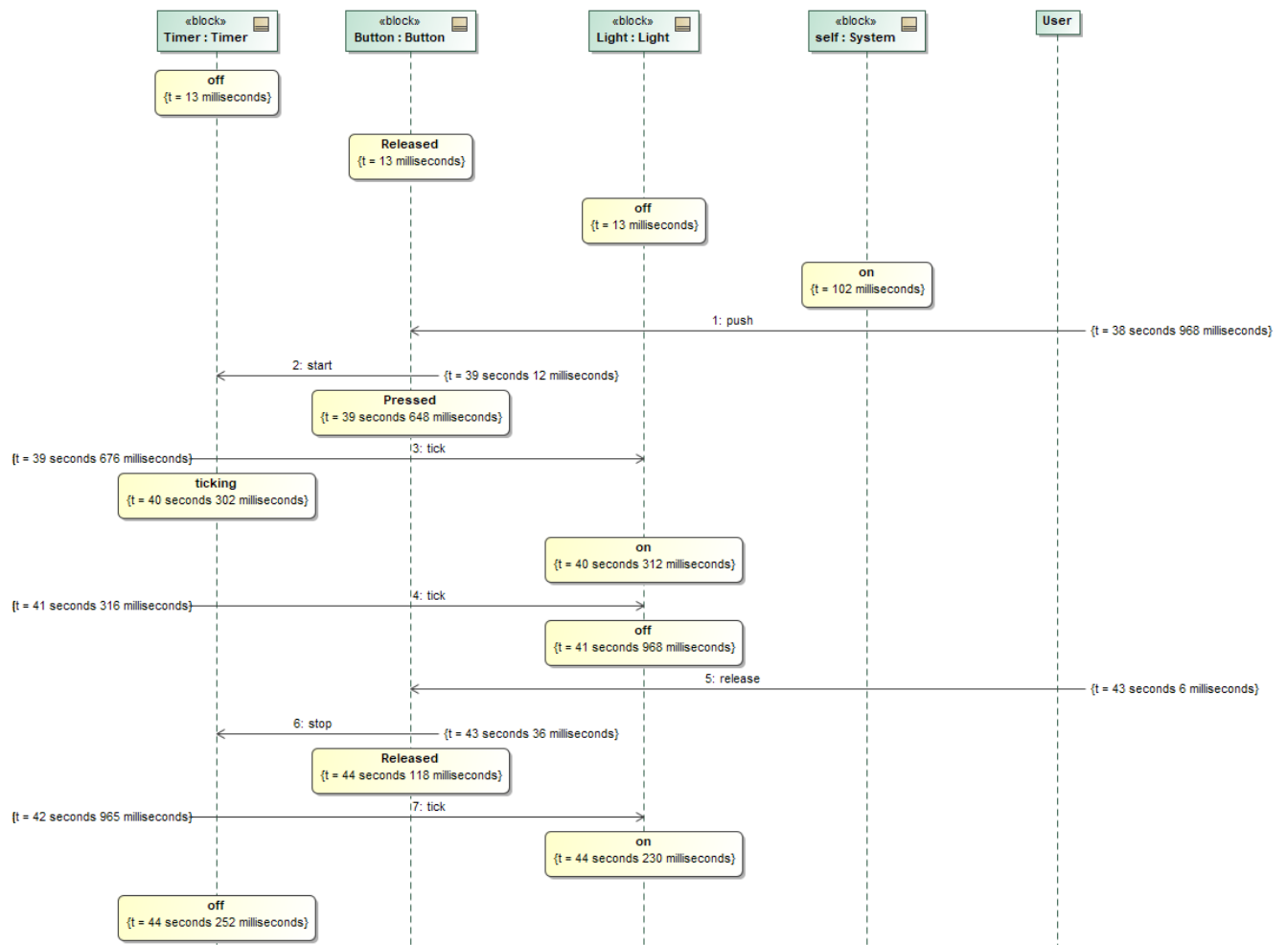
Whenever you simulate a model (such as Flashing Light as shown in the figure above), Magic Model Analyst will:

- create the first Lifeline, which represents the selected runtime object.
- record each signal sent from the selected runtime object as a Message in the sequence diagram.
- record each operation call caused by a call message, a CallOperationAction, or an ALH.callOperation with argument and return value as messages in the sequence diagram.
- record an object that receives a signal and/or an operation call as a Lifeline. Unless it exists in the diagram, the object will be called a 'lifeline object'.
- record each change in the state of a lifeline object as a StateInvariant on the Lifeline, with the changed state symbol.
- record each change in the feature value of a lifeline object as a StateInvariant on the Lifeline. Changes in value are enclosed in constraint brackets, i.e., {a=10}.



Note

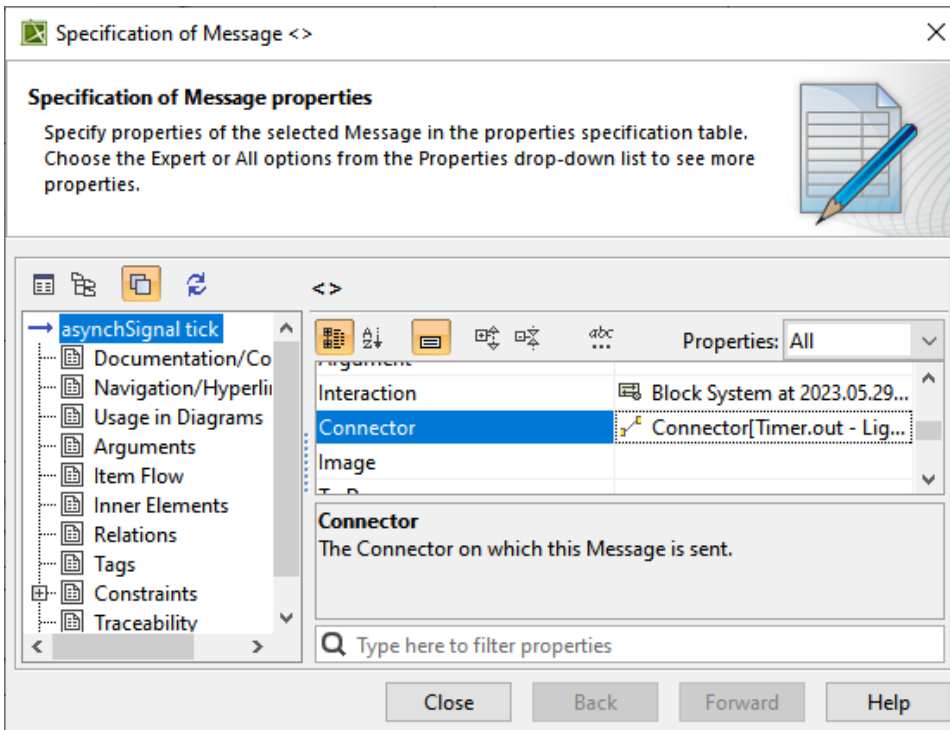
StateInvariants are designated by yellow rounded rectangles. See the following figure for examples.



A sequence diagram depicting call messages running between lifelines.

To see what connector a signal or an operation call is sent through

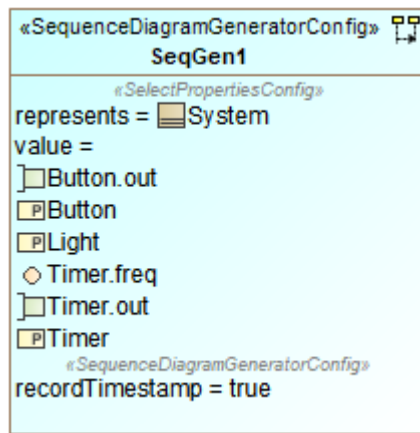
- Double-click the message or right-click it and select **Specification** to open the specification window.



To see the values of arguments sent with a signal or an operation call

1. Either double-click a message or right-click it and select **Specification** to open the specification window.
2. Select **Arguments** in the tree on the left-hand side of the specification dialog to see the value of each argument.

You can customize recorded messages (signals) and lifelines using **SequenceDiagramGeneratorConfig**.



A SequenceDiagramGeneratorConfig element showing its parameters.

A **SequenceDiagramGeneratorConfig** is a stereotype that is inherited from an **ExecutionListener** stereotype. It contains the following six tag definitions:

- **Owner**
An element that owns a generated Interaction element. A generated Sequence diagram will be created under that particular Interaction element. You need to select only the element that can own an Interaction element; otherwise, a model inconsistency will occur.
- **Ignored Signals**
Signals that will be ignored (will not be recorded) during a simulation recording.
- **Ignored Lifelines**
A list of elements (objects) that will be ignored (will not be recorded as lifelines) during a simulation recording. This list takes priority over the **value** list.

- **Recorded Object Path**

Used to specify a property path to an object that will be recorded by a sequence diagram generator. The path must start from a property owned by either a classifier which is the target of the simulation configuration, or a classifier of an instance specification which is the target of the simulation configuration. The property at each successive position following the first position must be contained in the classifier that types the property at the immediately preceding position.

- **Record State Change**

A boolean option. If true, state changes will be recorded.

- **Record Value Change**

A boolean option. If true, value changes will be recorded.

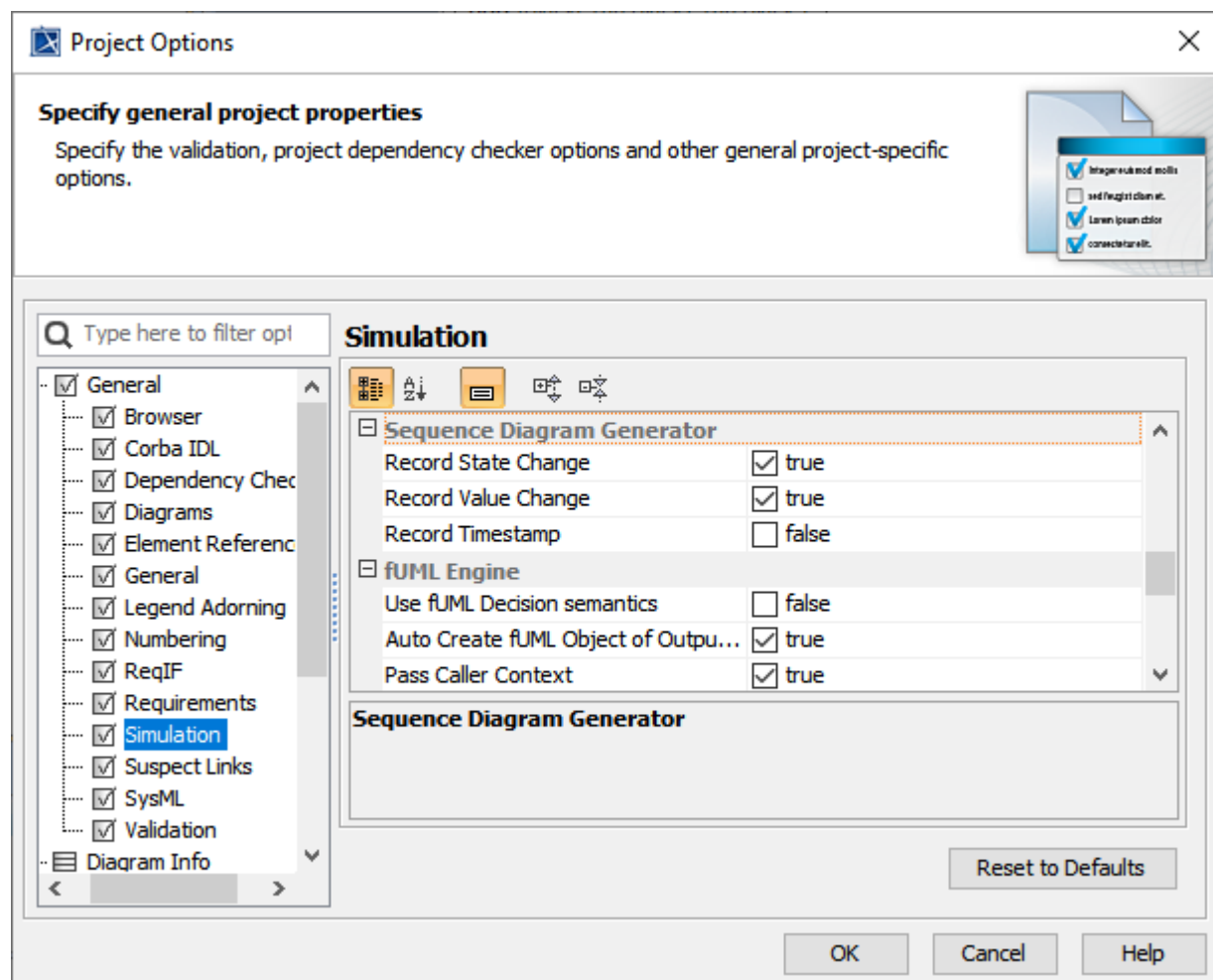
- **Record Timestamp**

A boolean option. If true, timestamps will be recorded on messages.

- **Value**

A structural feature whose value is represented by the configuration.

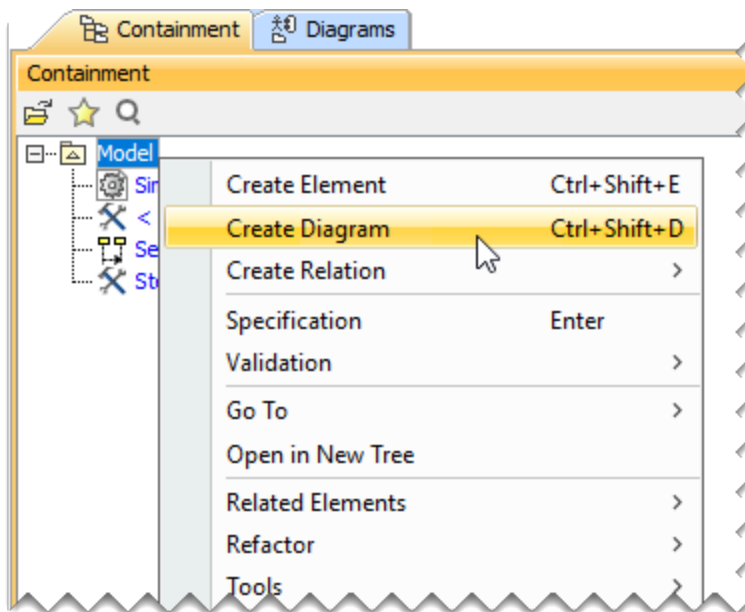
You can specify the default values of *recordStateChange*, *recordValueChange*, and *recordTimestamp* through the **Project Options** dialog. The values in the project options will be used if the tagged values of the sequence diagram generator are not specified. The default values of these project options are **true**, **true**, and **false** (see the following figure). **True** means all values will be recorded by default.



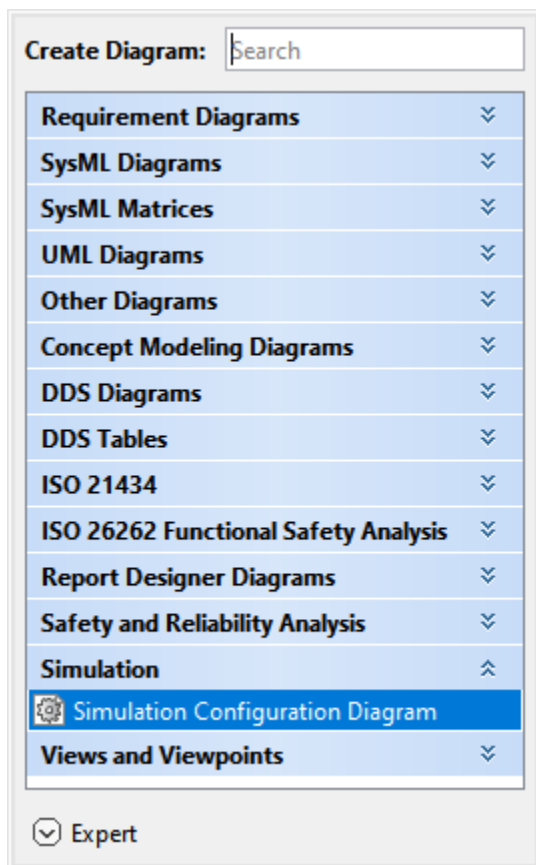
Options in the Sequence Diagram Generator group in the Project Options dialog.

To customize a Sequence diagram recording

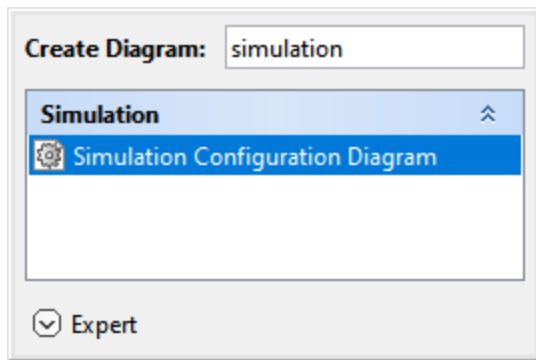
1. In the Containment tree, right-click the **Model** folder (root folder) and select **Create Diagram**.



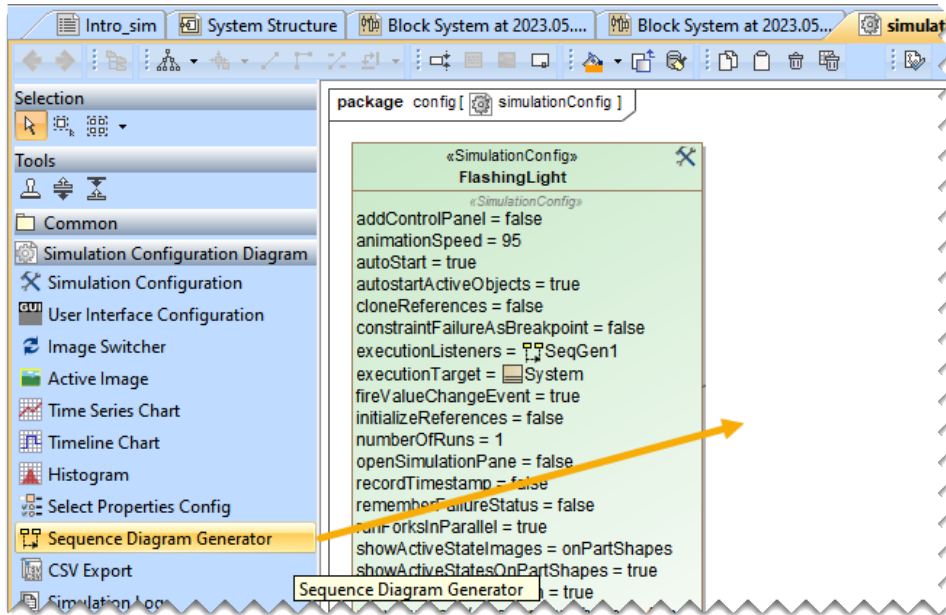
2. Do one of the following:
 - In the dialog, expand **Simulation** and select **Simulation Configuration Diagram**.



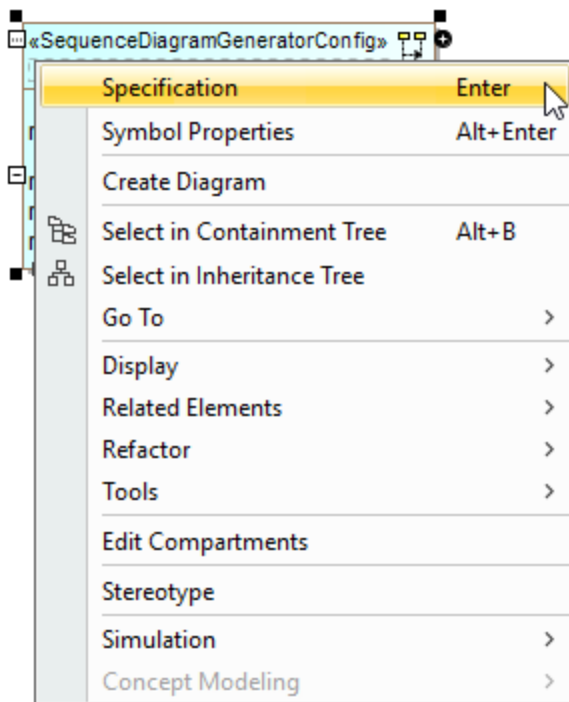
- In the search tab, type the keyword simulation and then select **Simulation Configuration Diagram**.



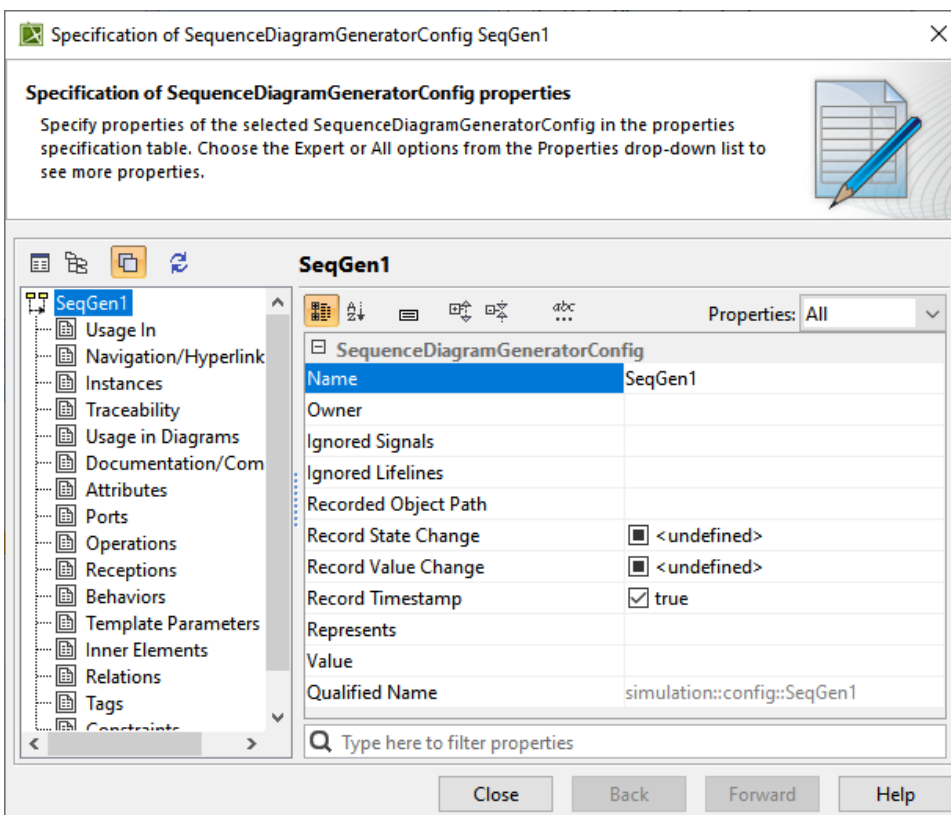
- From the **Simulation Configuration Diagram** palette, drag the **Sequence Diagram Generator** to the diagram pane.



- Right-click the newly created **SequenceDiagramGeneratorConfig** and select **Specification** to open its **Specification** dialog.

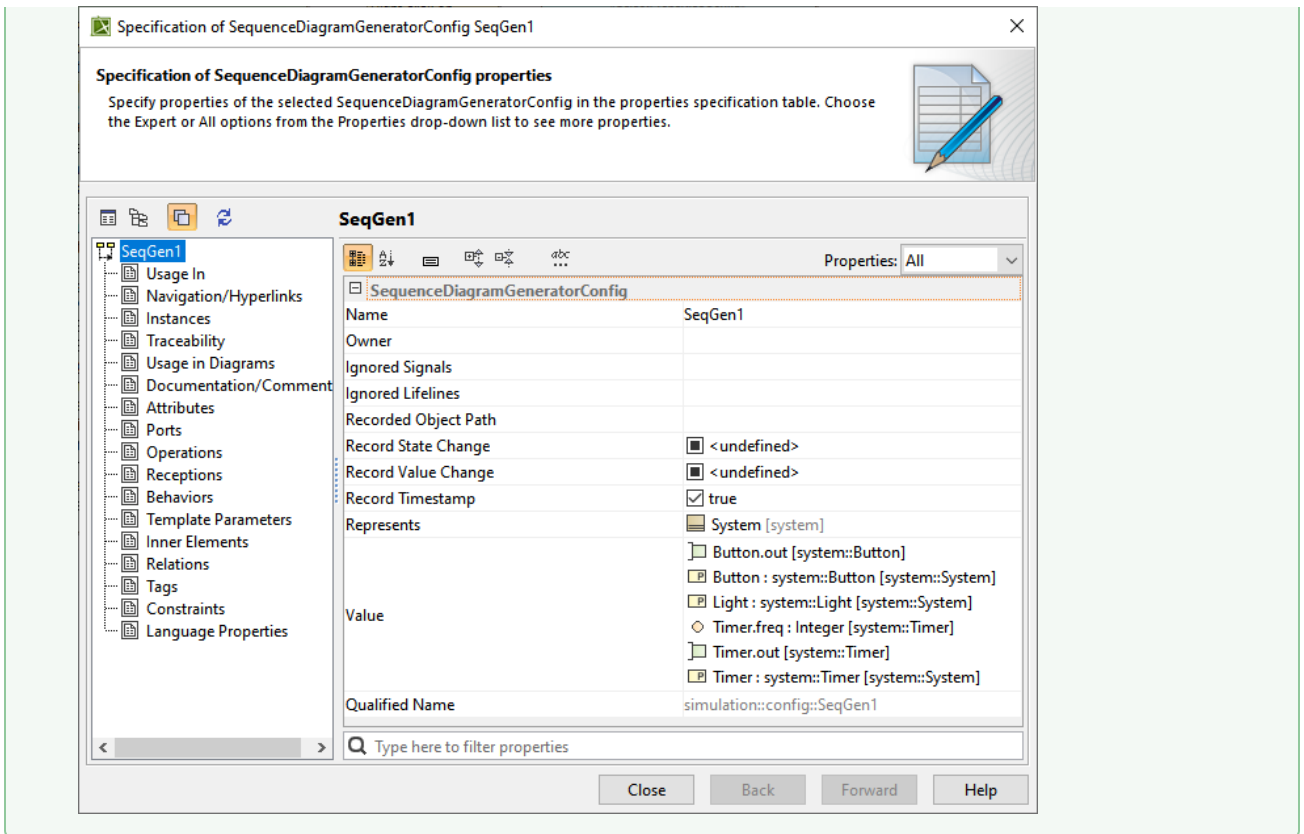


5. Specify the value(s) of the tag definition(s) of the config, e.g., **Name**.



Tip

You can specify **Represents** and **Value** to be captured by the **SequenceDiagramGeneratorConfig** along with other settings, such as **RecordTimestamp**.



6. Drag the configured **SequenceDiagramGeneratorConfig** to the **SimulationConfig** element (see [SimulationConfig stereotype](#)). The **executionListeners** tag of the **SimulationConfig** will be shown with the specified name of the **SequenceDiagramGeneratorConfig**, as shown below.

