# State Machine simulation

Cameo Simulation Toolkit allows you to perform a State Machine Simulation (State Chart Simulation) on existing State Machine diagrams, based on the W3C SCXML standard. This kind of simulation is frequently used in the early stage of software development by designers or analysts to test the flow of the software to be developed.

The W3C SCXML standard provides a generic State machine-based simulation environment based on the Harel Statechart. SCXML is capable of describing complex State machines, including Substates, concurrency, history, time events, and many more. Most of the things that can be represented as UML statecharts, e.g., business process flows, views on navigation bits, interaction or dialog management, and many more, can leverage the SCXML engine. When executing a State Machine, the SCXML engine is capable of finding an initial state automatically even if the initial node is not defined. This feature is also applicable to composite States and orthogonal States.

In addition to simulating an executable model as a demonstration tool to validate and verify the system Behavior at key milestone reviews, the State Machine simulation supports exporting the UML State Machine to standard SCXML files for further analyses or transformations.

To export the UML State Machine to an SCXML file

1. Open a Simulation project with the State Machine diagram.
2. From the main menu, select **File** > **Export To** > **SCXML File**. The **SCXML File** dialog opens to select the State Machine diagram exported to an SCXML file.
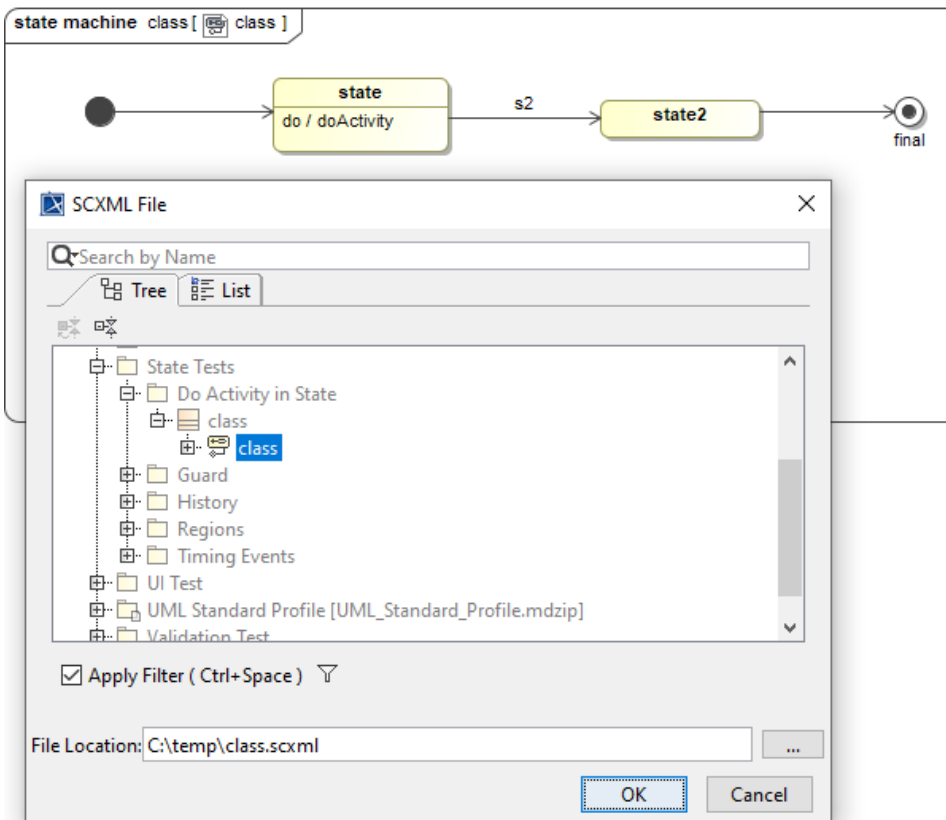
3. Locate the State Machine diagram by doing one of the following:
   - In the *Search by Name* box, type the name of the State Machine diagram.
   - Click the **Tree** tab and select the State Machine diagram appearing in the tree.
   - Click the **List** tab and select the State Machine diagram appearing in the list.

   - At the **File Location** box, select ... to browse the State Machine diagram.
4. Click **OK**. The exported file, e.g., *class.scxml*, will be as follows.

```
<scxml xmlns="http://www.w3.org/2005/07/scxml" version="null" initial="
initial__20_0_5630196_1596545104923_661917_23868">
        <!-- http://commons.apache.org/scxml -->
        <state id="initial__20_0_5630196_1596545104923_661917_23868">
                <transition target="State_Tests::Do_Activity_in_State::class::class::
_20_0_5630196_1596545104918_849035_23846::state">
                </transition>
        </state>
        <state id="State_Tests::Do_Activity_in_State::class::class::
_20_0_5630196_1596545104918_849035_23846::state">
                <transition event="s2" target="State_Tests::Do_Activity_in_State::class::class::
_20_0_5630196_1596545104918_849035_23846::state2">
                </transition>
                <invoke targettype="doactivity" src="_20_0_5630196_1596545104909_691815_23820">
                </invoke>
        </state>
        <state id="final_State_Tests::Do_Activity_in_State::class::class::
_20_0_5630196_1596545104918_849035_23846::final" final="true">
        </state>
        <state id="State_Tests::Do_Activity_in_State::class::class::
_20_0_5630196_1596545104918_849035_23846::state2">
                <transition event="COMPLETION_EVENT__20_0_5630196_1596545104923_763628_23871" target="
final_State_Tests::Do_Activity_in_State::class::class::_20_0_5630196_1596545104918_849035_23846::final">
                </transition>
        </state>
</scxml>
```

Related pages

- Supported elements
- Adapting models for State Machine simulation
- Running a State Machine simulation
- State Machine duration simulation
- Sample projects