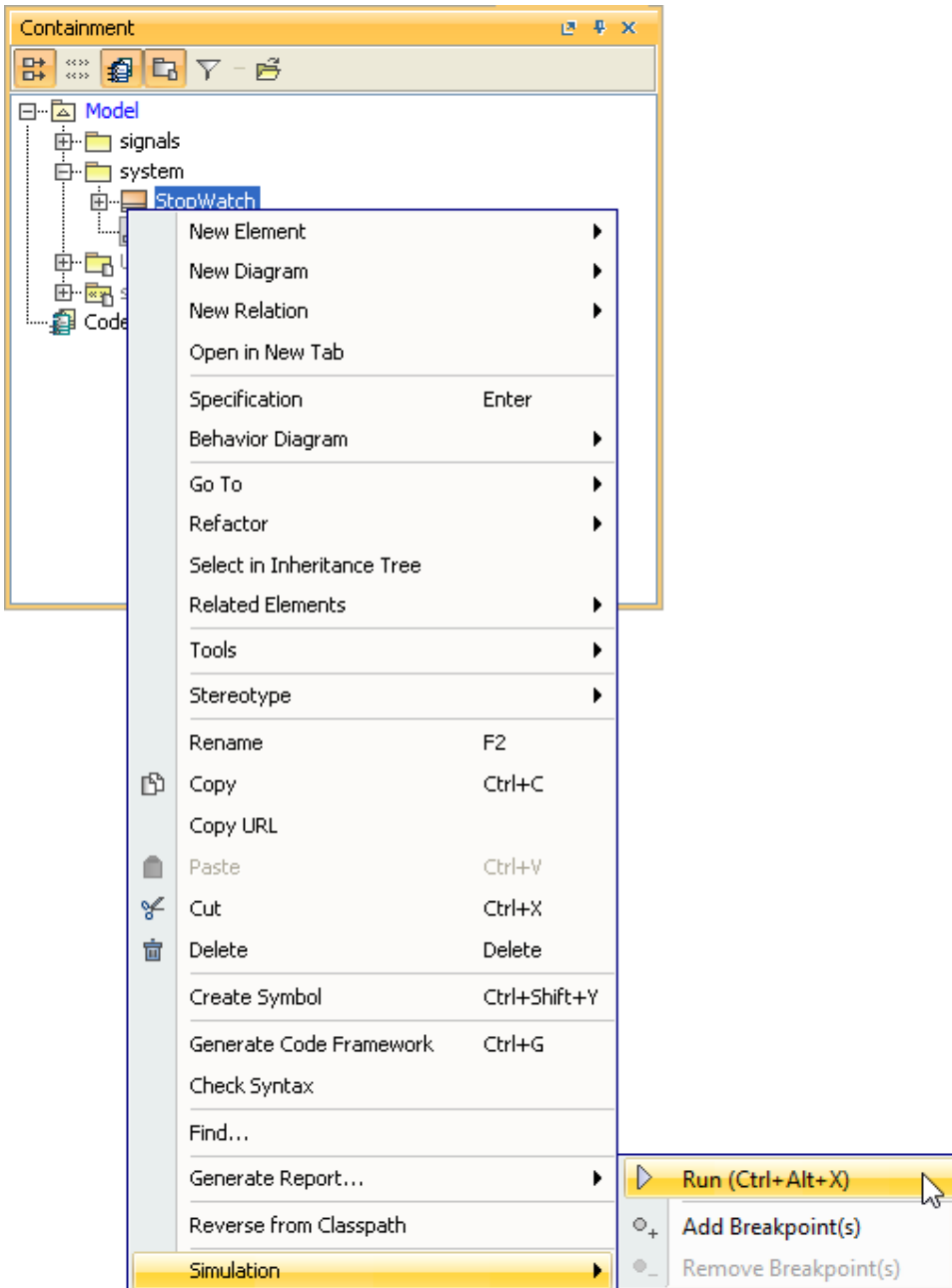


# Executing the Stopwatch class

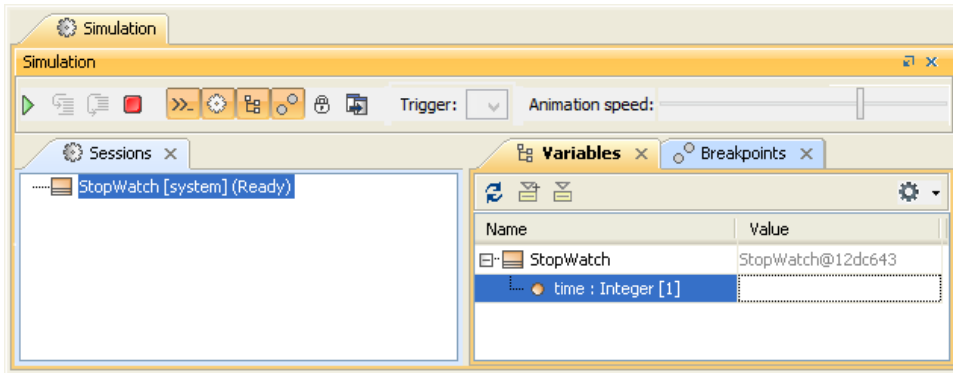
The Stopwatch Class is now ready to be executed. You can execute the Stopwatch Class either from the containment browser or from the diagram.

To execute the Stopwatch Class

1. Either (i) select the **StopWatch** class in the containment tree and click the **Run** button on the **Simulation Control** main toolbar or (ii) right-click the **StopWatch** class and select **Simulation > Run**.



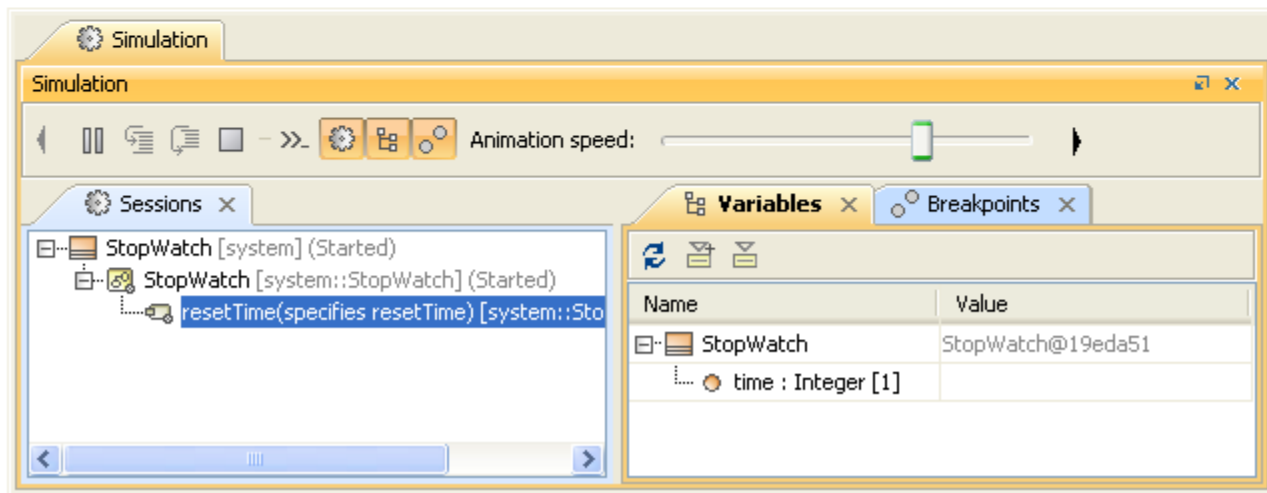
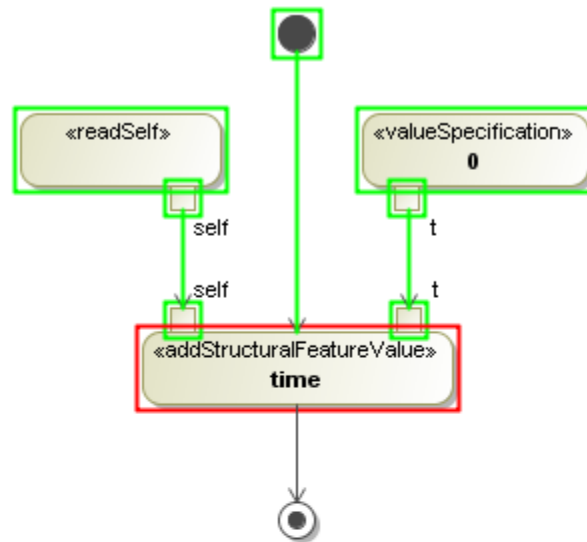
2. The **Simulation** window will open. It contains four tabs: (i) **Sessions**, (ii) **Console**, (iii) **Variables**, and (iv) **Breakpoints** tabs. At this point, a simulation session to execute the **StopWatch** Class will be created. You can see this session in the **Sessions** tab of the **Simulation** window.



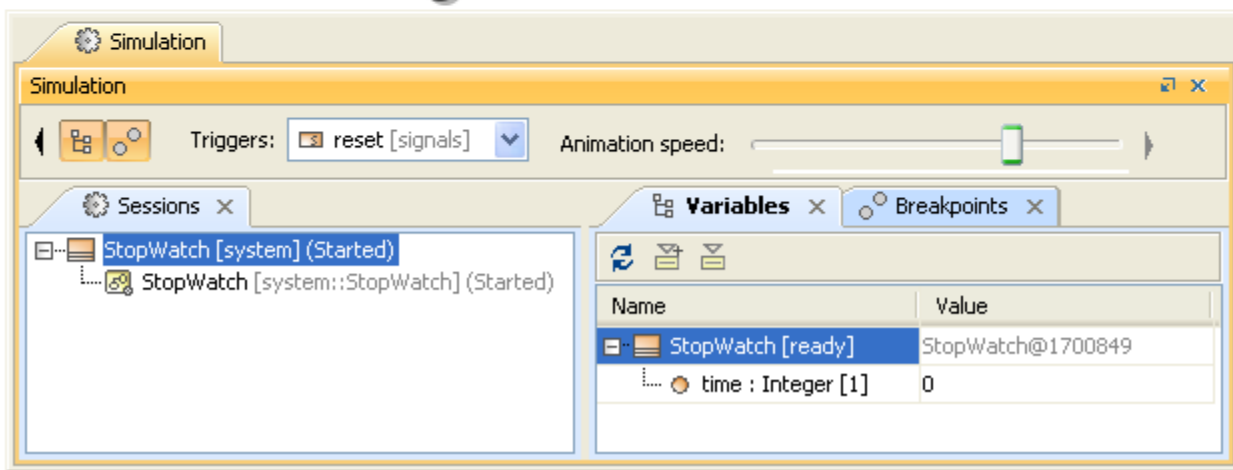
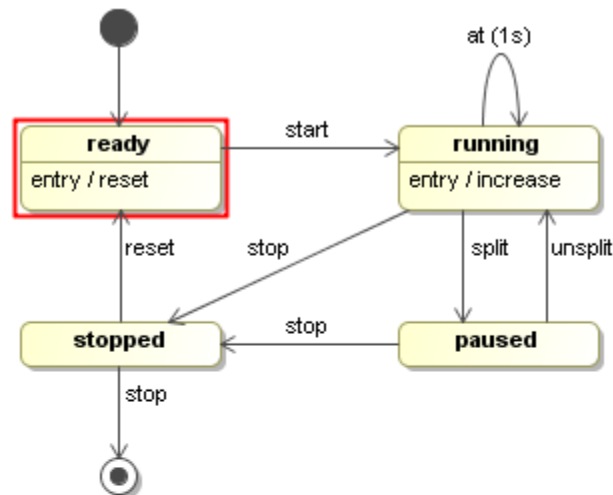
3. Click the **Start** button  on the **Simulation** window toolbar.

Once you have clicked the **Run Execution** button, the Object of the **StopWatch** Class will be created and shown in the **Variables** tab. The Classifier Behavior of the **StopWatch** class and the **StopWatch** State Machine will be executed respectively. A new session to execute the **StopWatch** State Machine will be created under the simulation session of the **StopWatch** class, and the **StopWatch** State Machine diagram will be open and ready for the simulation.

The **StopWatch** State Machine execution will start from the initial stage and automatically move to the **ready** state because the transition which connects between the initial stage and **ready** state does not have a defined trigger. When the **ready** State is entered, the **reset** Activity, which is assigned as the entry to the **ready** State, will be invoked. This Activity also calls the **resetTime** activity of the StopWatch Class for setting the time value to be zero. At this point, a new simulation session will be created to execute the **resetTime** Activity, and the **resetTime** Activity diagram will be open and animated.



The execution of the **resetTime** Activity will be terminated once it has reached the final stage. When the simulation stops, the session executing the **resetTime** Activity will be closed, the StopWatch State Machine diagram will be reactivated, and the simulation will wait for an Event signal at the ready State. You can browse the **StopWatch** object in the **Variables** tab to see the time value.



Now we want to trigger the Stopwatch object from the ready State to the running State. To do this, we need to send a start signal to the Stopwatch object.

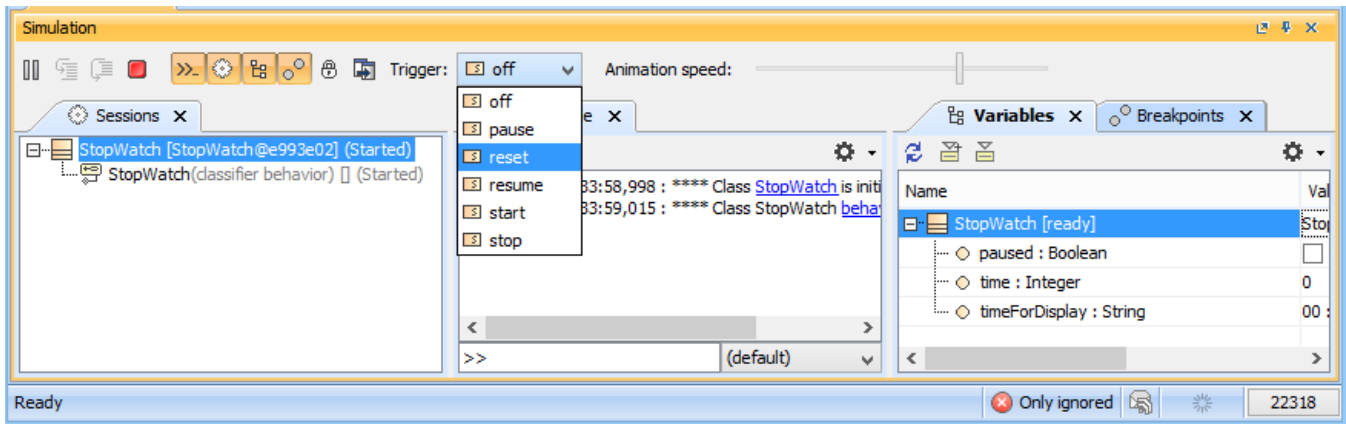
To send an event (start) signal to the object of the **StopWatch** Class

1. Select the **StopWatch** node in the **Variables** tab. The signals that can be sent to the **StopWatch** will be listed in the **Trigger** drop-down menu.
2. Select the **start** signal from the **Triggers** drop-down menu (see the following figure).

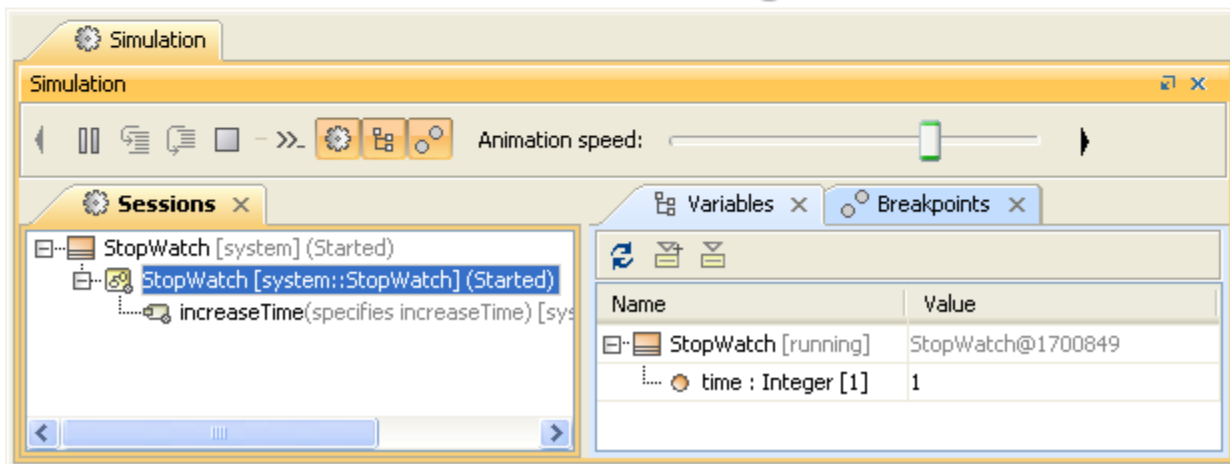
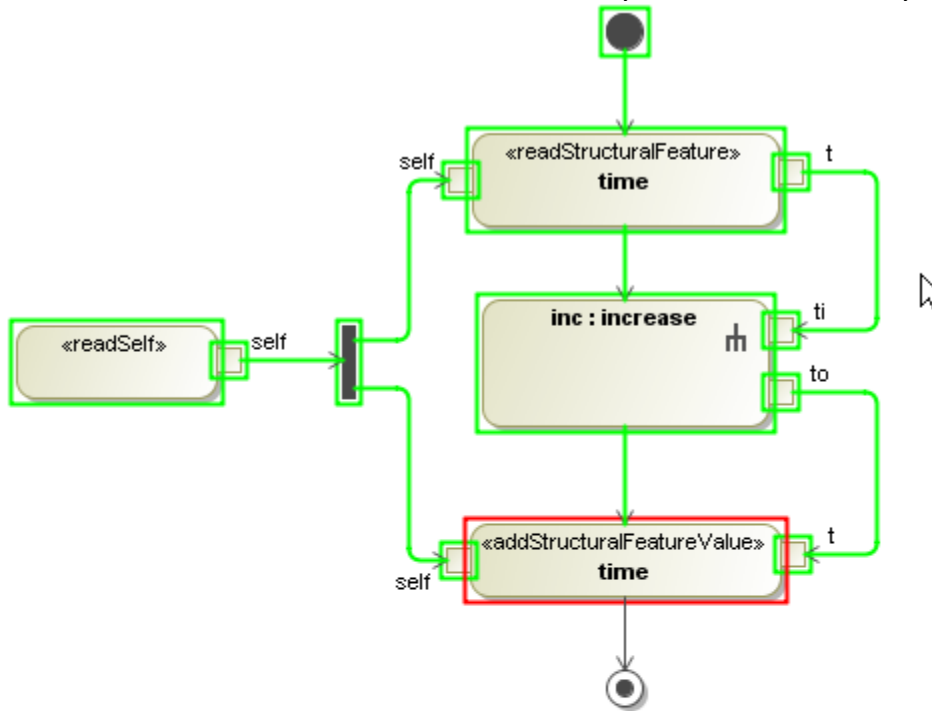


#### Note

You can click the arrow button on the left or right-hand side of the **Simulation** window toolbar to scroll the toolbar to the left or to the right.

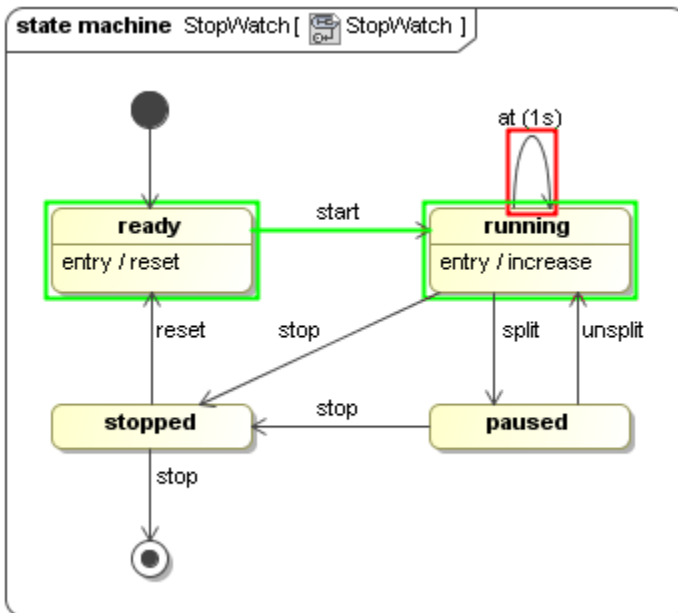


Once the start Signal has been sent, the **StopWatch** will go to the **running** State. At the entry to the **running** State, the **increase** Activity will be defined; therefore, it will be invoked. The **increase** Activity will then call the **increaseTime** Activity of the **StopWatch** Class to increment the time value by one second. A new simulation session will be created to execute the **increaseTime** Activity, and the **increaseTime** Activity diagram will be open and animated.



The **increaseTime** Activity will be completely executed when its Final stage has been reached. Once the execution is complete, the simulation session of the **increaseTime** Activity will be closed and the **StopWatch** State Machine will be activated again. Now you can see in the **Variables** tab that the time value of the Stopwatch object has been incremented by one.

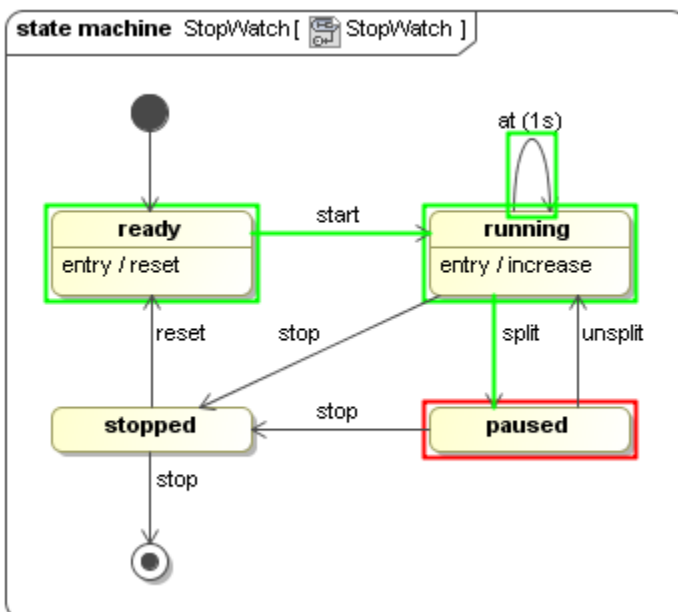
The Stopwatch object will still be at the **running** State and the time Event will be triggered at this State every second. Therefore, a new simulation session will be created to execute the **increaseTime** activity every second, and the time value will increment by one every time the **increaseTime** Activity is executed.



While the Stopwatch object is at the **running** State, we can send a **split** signal to trigger the object to the **paused** State or send a **stop** signal to trigger the object to the **stopped** State.

To send a split signal to the Stopwatch object

1. Select the Stopwatch object in the **Variables** tab.
2. Select the **split** signal from the **Triggers** drop-down menu on the **Simulation** window toolbar. The state of the Stopwatch object will be changed to the **paused** State.

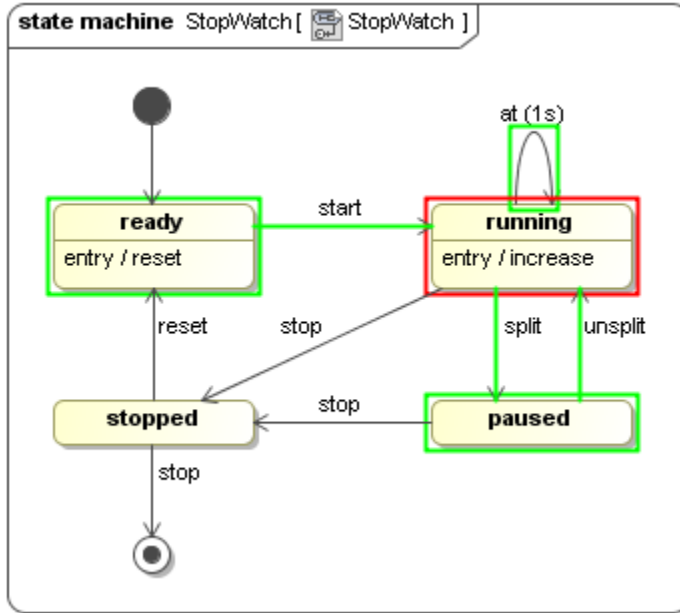


You can change the paused Stopwatch object to the **running** State by sending an **unsplit** signal to the object.

To change the Stopwatch object from a paused state to a running State

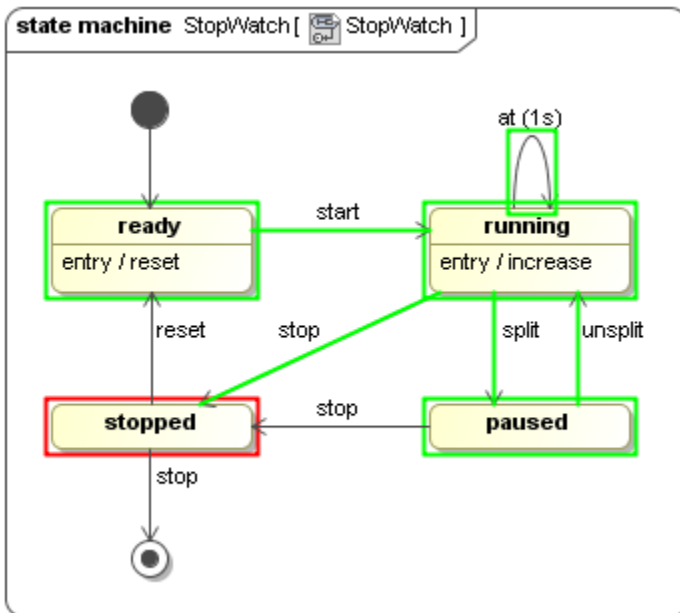
1. Select the Stopwatch object in the **Variables** tab.

2. Select the **unsplit** signal from the **Triggers** drop-down menu on the **Simulation** window toolbar. The state of the Stopwatch object will be changed to the **running** State, and the time value will continuously increase.



To send a stop Signal to the Stopwatch object

1. Select the Stopwatch object in the **Variables** tab.
2. Select the **stop** signal from the **Triggers** drop-down menu on the **Simulation** window toolbar. The State of the Stopwatch object will be changed to the **stopped** State.



At the **stopped** State, if the **reset** signal is sent to the Stopwatch object, the object will go to the **ready** State, and the time value will be reset to zero at the entry to the **ready** state by calling the **resetTime** Activity. If the **stop** signal is sent to the object again, the object will go to the final stage, and then the simulation will stop and all of the simulation sessions will be closed.