Manipulating runtime objects using scripts

In Magic Model Analyst, you can use the following types of action scripts to manipulate runtime objects:

- fUML object syntax, which simplifies some ALH actions. Example: object.get("property")
- ALH (Action Language Helper) scripts. Example: ALH.getValue(object, "property")
- Alf - Action Language for fUML (requires an additional plugin). Example: object.property

Action scripts can be used anywhere where scripts are evaluated during simulation (e.g., Opaque Behaviors, Actions, Guards, or even directly in the simulation **Console** pane). You can use them to get and specify a Structural Feature value, call a specific behavior or operation, create a runtime object, get its current State, create a Signal instance, and more.

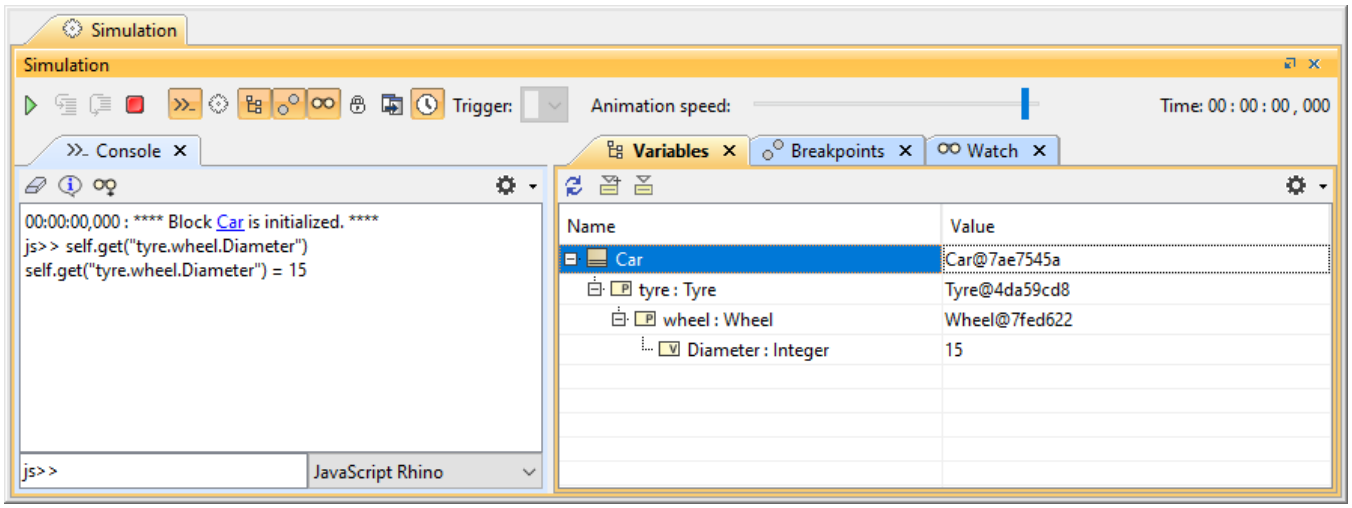
> ⚠ Note that scripts depend on the objects where they are executed, i.e., scripts are context-dependent.

When executing a script, you always need to specify its context (object). Most commonly, it is the current simulation context (marked as "self" in a script). However, it can also be an input pin (pin name) or an object created in a script itself (see Creating a runtime object).

Analyze the example described below to see how scripts change depending on the context.

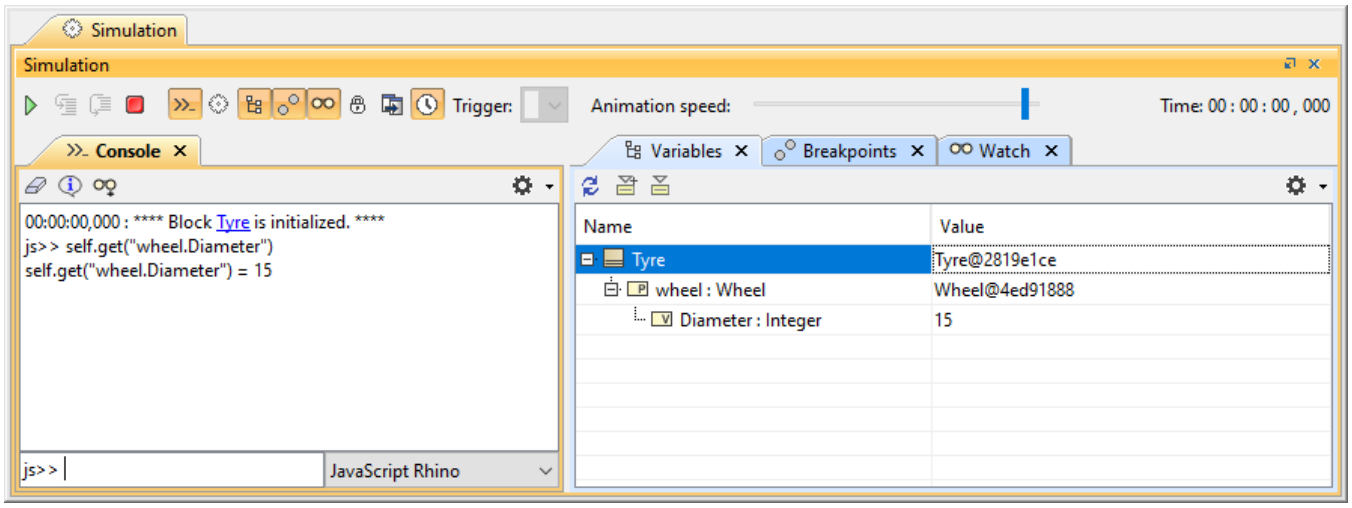
Example

In the example below, to get the *Diameter* Value Property when the script is executed in the *Car* Block, the script should be **self.get("tyre.wheel.Diameter")**.



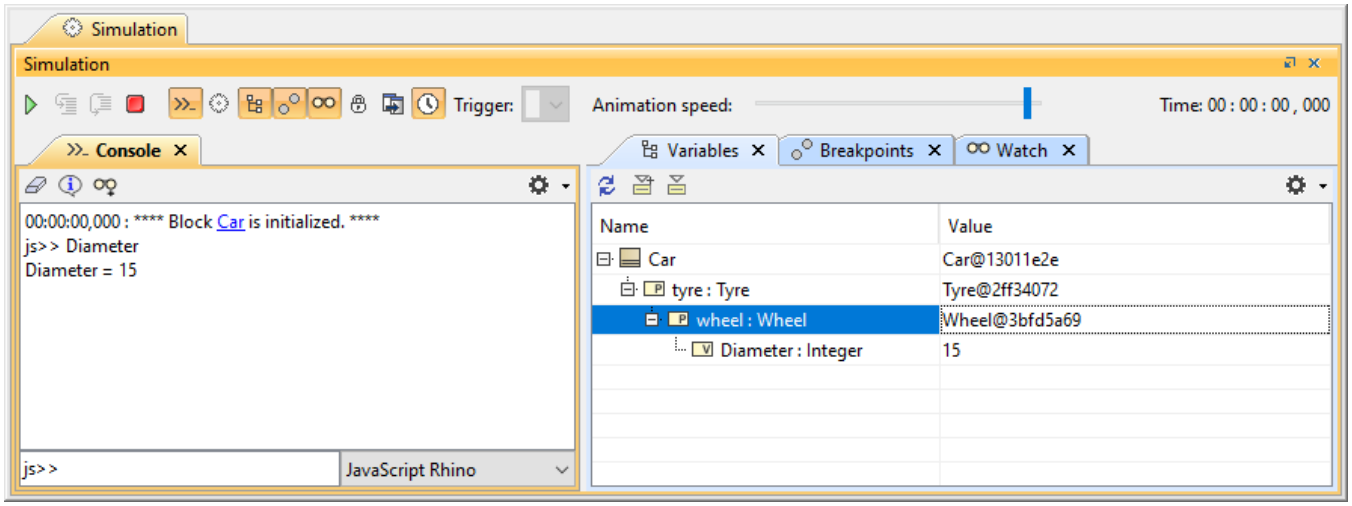
Getting the *Diameter* Value Property when the simulation context is the *Car* Block.

However, if you wanted to get the same *Diameter* Value Property when executing the script in the *Tyre* Block, the script should be **self.get("wheel.Diameter")**.



Getting the *Diameter* Value Property when the simulation context is the *Tire* Block.

Note that when executing scripts directly in the simulation **Console** pane, the context is the currently selected element in the **Variables** pane. So even if the simulation context is the *Car* Block, selecting the *wheel* Part Property in the **Variables** pane allows you to access the *Diameter* Value Property directly, as shown below.



Getting the *Diameter* Value Property when the script is executed in the Console pane.

Tip

It is easy to access elements that are deeper in the hierarchy tree, but accessing parent elements is more complicated. Therefore, it is important to plan where executing a particular script would be the most efficient.

To learn more, refer to the following topics:

- Manipulating structural values
- Manipulating global variables
- Calling behaviors and operations
- Creating a runtime object
- Creating and sending signals
- Getting a token value
- Checking States
- Evaluating an expression
- Creating an ArrayList in Java
- Getting simulation context
- Getting simulation time
- Getting the caller of a script

Additional resources

- You can find more examples of API usage at *<install_root>\samples\simulation\SmallTestSamples.mdzip* (Tests section).
- For more information, see the JavaDoc at *<install_root>\openapidocs\SimulationJavaDoc*.