

# Action scripts APIs

Magic Model Analyst uses Java Scripting API to evaluate scripting languages. It is a scripting language-independent framework that uses script engines from Java code.

Magic Model Analyst supports the following scripting languages

- ECMLScript/Javascript (default)
- Beanshell
- Groovy
- Python
- Ruby

You can use the supported scripting languages in the UML elements, e.g., OpaqueExpression and OpaqueBehavior. The available Helper Classes that you can use with those scripting languages are the ALH (Action Language Helper) Class and ExcelHelper. For more information, see the JavaDoc of ALH in the MagicDraw installation directory (*openapi/docs/SimulationJavaDoc*) and ExcelHelper Class.

This section explains the existing predefined variables of Action Scripts and demonstrates the use of ALH scripting with the following APIs

- Getting a Structural Feature Value
- Specifying a Structural Feature Value
- Calling a Specific Behavior
- Calling a Specific Operation
- Creating a Run-time Object
- Creating a Signal Instance
- Sending a Signal Instance to a Specific Target Object
- Getting a Token Value
- Getting the Current State of a Run-time Object
- Getting the Last Signal Instance from a Run-time Object
- Evaluating an Expression
- Creating an ArrayList in Java
- Checking the State of an Object
- Adding a Value to an Object
- Removing a Value of an Object
- Getting a Context
- Accessing Current Simulation Time
- Accessing the Simulation Time Unit
- Adding a Value to a Global Variable
- Getting a Value from a Global Variable
- Removing a Defined Global Variable
- Checking an Existing Global Variable

You can find more examples of the use of ALH in *<MagicDraw installation directory>/samples/simulation/tests/test\_ALH.mdzip*. Furthermore, most ALH APIs interact with the fUML abstract syntax metaclasses (see <http://www.omg.org/spec/FUML/>).