

Executable concepts

We call executable such concepts that can be specified by some expressions and calculated. In modeling tools developed by No Magic, executable are:

- [Validation rules](#) - that are predefined or custom constraints having a purpose to validate a model or particular model elements.
- [Derived properties](#) - properties that are automatically calculated from the other properties.
- [Smart packages](#) - that are special collections of model elements.
- [Executable opaque behaviors](#) - reusable expression fragments callable from other executable concepts.
- [Metric definitions](#) - calculate aggregate parameters of models.
- [Macros](#) - a script that is created using a particular script language. With the help of macros, you can control items that are allowed in [Developer Guide](#).
- [Opaque behaviors](#) - a behavior with implementation-specific semantics.

In a modeling tool, executable concepts are defined in one of the languages that our modeling tool can evaluate:

- [StructuredExpression](#) - a graphical way to construct expressions out of predefined blocks.
- [Object Constraint Language](#).
- [Binary](#) - an expression that is written in Java, compiled and added to our modeling tool with a plugin.
- One of the following scripting languages:
 - Groovy
 - JRuby
 - JavaScript
 - JavaScript Rhino
 - Jython



Additional scripting languages

Additional scripting languages can be plugged into our modeling tool, if they have JSR-223-compliant implementation.

Related pages

- [Object Constraint Language](#)
- [Binary](#)

You can test and evaluate your expressions with the help of the [Expression Evaluation tool](#).