

Creating validation rules

Validation rules and validation suites specify what will be validating and how. They also specify how a problem that is found by a validation rule can be solved.

A **validation rule** is a constraint with the applied stereotype «UML Standard Profile::Validation Profile::validationRule». Validation rules can validate model elements and non model elements (for example, presentation elements) as well. The UML metaclass specified as a constrained element defines the validation rule that validates elements of the specified metaclass. The stereotype specified as a constrained element specifies the validation rule that validates elements having the stereotype applied. A classifier specified as a constrained element specifies the validation rule that validates instances of the specified classifier. Validation rule's implementation can be OCL2.0 based or binary. Binary validation rules can be implemented in Java. OCL2.0 based validation rules are described using the OCL2.0 language. The validation rule can be global or local one. Global validation rules are executed only once per validation session. Local validation rules are executed for each model element. The validation rule is treated as global if:

- it is the OCL2.0 based validation rule, the OCL2.0 specification is valid, and the OCL2.0 specification does not use the "self" variable (explicitly or implicitly by using the property of a constrained element only)
- it is binary based, has the specified implementation class name, and it has no specified constrained elements.

A **validation suite** is a package with the stereotype «UML Standard Profile::Validation Profile::validationSuite». The validation suite organizes several validation rules into a unit that can be used for performing the validation. Validation rules can be added or imported into the validation suite.

An **active validation suite** is a package with the stereotype «UML Standard Profile::Validation Profile::activeValidationSuite». Active validation rules can be checked constantly or on the model elements change. OCL2.0 or binary validation rules can be used in the active validation. We suggest to prefer binary, because they give the better performance.

An **annotation** defines the validation result. It contains information about what is incorrect, severity of the problem, and possible actions that can be used to solve the problem.

Related pages

- [Annotating the elements](#)
- [Validation rule developer's roadmap](#)
- [Create OCL2.0 validation rule](#)
- [Binary validation rule](#)
- [Implementing a binary validation rule](#)
- [Binary validation rule in the plugin](#)
- [Script validation rule](#)
- [How to provide a solution for a problem found during validation?](#)