

# Binary validation rule

Sometimes it is hard to specify the rule for some advanced concepts in OCL. For example, the string manipulation is rather weak in OCL2.0, hence the writing rule for string formats, parsing strings is very hard in OCL. In such cases the modeling tool offers possibility to write rules in Java language and invoke them when validating.

Common steps for creating a binary validation rule:

1. Create a constraint
2. Set the stereotype «UML Standard Profile::Validation Profile::validationRule» for the validation rule
3. Set the severity level, error message, and abbreviation
4. Specify constrained element(s)
5. Specify the specification language binary
6. Create a Java implementation class (see all possible [cases](#))
7. Compile Java code, resulting in one or several classes.
8. Ensure that your modeling tool can find & load these classes. You can place them in the program classpath or use the modeling tool plugins mechanism. The same rules as writing the code for the open API are used here.

These steps are common to all possible binary validation cases and will not be repeated in each case description.

See the *MyBinaryValidationRuleImpl.java* example in the *<program installation directory>\openapi\examples\validation* directory.



The implementation class must be in the classpath of the program or if the implementation class is in a plugin then the plugin's classloader must be registered to the validation system (see [Binary validation rule in the plugin](#)).

## Related pages

- [Creating validation rules](#)