Invoking Variant Realization transformation from command line

To invoke the variant realization transformation, a command line utility is available at *<install.dir>/plugins/com.nomagic.magicdraw.variabilitycore/transform. bat.*

The *transform.sh* for Unix (including Apple) installation file is also located in the same directory.

This command line utility accepts multiple options. To execute it, use the following syntax:

```
transform.bat -option1 value1 -option2 value2 -option3 value3
```

In the following table, you can find the available options:

Option	Description
-project	A project to work on. It can be either local project or project on the server.
-scope	A scope of variant realization transformation: (a qualified name of) a package.
-variantinstance	A variant configuration to be realized. A qualified name of the instance specification.
-savevariantas	**[For local projects]** Save the result project into this file.
-server_url	**[For server projects]** Server URL to connect to.
-username	**[For server projects]** User name.
-password	**[For server projects]** User password.
-outputbranch	**[For server projects, output alternative 1]** Commit result project into this branch.
-commitmessage	**[For server projects, output alternative 1]** Provide a commit message for committed variant project.
-outputdir	**[For server projects, output alternative 2]** Save project as file into this directory.
-workdir	A working directory. This directory is used as a base for specifying project, *savevariantas*, and *outputdir*.
-ignorelocks	This setting is equivalent to the GUI option **Forcibly Transform Read Only Elements** of the Variant Realization transformation.
-keepconstraints	This setting is equivalent (but opposite) to the GUI option **Clean Variation Point Data** of the Variant Realization transformation.
-retainversions	**[For server projects, output alternative 1]** This is a quite complicated option. It governs the used projects of the project being processed. When committing the result variant project into the branch, this option makes it so that used module versions remain as they were in the branch. Without this option, used module versions will be as they are in the project being transformed.

- properties	A file to load options from. All the aforementioned options can be specified in a file using Java properties file format. Basically, a list of properties in the form: option1=value1 option2=value2 For more details about the format, see Java documentation at https://docs.oracle.com/javase/9/docs/api/java/util/Properties.html. Note, that it is possible to mix the use of both mechanisms. You can specify a half of the options on the command line and the other half in the properties file.