



Maven Plugin Guide

OpenL Tablets BRMS

Release 5.21





OpenL Tablets Documentation is licensed under a [Creative Commons Attribution 3.0 United States License](https://creativecommons.org/licenses/by/3.0/us/).

Table of Contents

1	Preface.....	4
1.1	Related Information	4
1.2	Typographic Conventions	4
2	Introduction	5
2.1	Goals Overview.....	5
2.2	Usage Overview	5
3	Goals.....	7
3.1	openl:generate	7
	Required Parameters	7
	Optional Parameters	7
	Parameter Details	8
3.2	openl:compile.....	12
	Required Parameters	12
	Optional Parameters	13
	Parameter Details	13
3.3	openl:test	13
	Required Parameters	14
	Optional Parameters	14
	Parameter Details	15
3.4	openl:package	16
	Required Parameters	17
	Optional Parameters	17
	Parameter Details	17
3.5	openl:deploy.....	19
	Required Parameters	19
	Optional Parameters	19
	Parameter Details	20
3.6	openl:help.....	21
	Optional Parameters	21
	Parameter Details	21
4	Usage	23
4.1	Directory Structure.....	23
4.2	Integrate openl-maven-plugin in the Maven Lifecycle	23
4.3	Configure Interface, Domain Classes and Project Descriptor Generation	24
4.4	Configure OpenL Project Compilation and Validation	24
4.5	Configure OpenL Project Testing.....	25
5	Examples.....	26
5.1	Configuration with all OpenL Maven Plugin Goals.....	26
5.2	Creating a Project with a Working Example of OpenL Maven Plugin Usage.....	27

1 Preface

This preface is an introduction to the *OpenL Tablets Maven Plugin Guide*.

The following topics are included in this preface:

- [Related Information](#)
- [Typographic Conventions](#)

1.1 Related Information

The following table lists sources of information related to contents of this guide:

Related information	
Title	Description
http://openl-tablets.org/	OpenL Tablets home page.

1.2 Typographic Conventions

The following styles and conventions are used in this guide:

Typographic styles and conventions	
Convention	Description
Bold	<ul style="list-style-type: none"> • Represents user interface items such as check boxes, command buttons, dialog boxes, drop-down list values, field names, menu commands, menus, option buttons, perspectives, tabs, tooltip labels, tree elements, views, and windows. • Represents keys, such as F9 or CTRL+A. • Represents a term the first time it is defined.
Courier	Represents file and directory names, code, system messages, and command-line commands.
Courier Bold	Represents emphasized text in code.
Select File > Save As	Represents a command to perform, such as opening the File menu and selecting Save As .
<i>Italic</i>	<ul style="list-style-type: none"> • Represents any information to be entered in a field. • Represents documentation titles.
< >	Represents placeholder values to be substituted with user specific values.
Hyperlink	Represents a hyperlink. Clicking a hyperlink displays the information topic or external source.
<i>[name of guide]</i>	Reference to another guide that contains additional information on a specific feature.

2 Introduction

Access to rules and data in Excel tables is realized through OpenL Tablets API. OpenL Tablets provides wrappers to developers to facilitate easier usage.

This plugin is used to generate interface to access the rules, to validate rules during compilation phase, and to run OpenL Tablets tests.

The following topics are included in this section:

- [Goals Overview](#)
- [Usage Overview](#)

To acquire a better understanding of plugin name usage, see the following examples:

- [Configuration with all OpenL Maven Plugin Goals](#)
- [Creating a Project with a Working Example of OpenL Maven Plugin Usage](#)

2.1 Goals Overview

General information about the goals is as follows:

General information about the goals	
Link to the goal section	Description
openl:generate	Generates OpenL Tablets interface, domain classes, and project descriptor.
openl:compile	Compiles the OpenL Tablets project.
openl:test	Runs OpenL Tablets tests.
openl:package	Packages the OpenL Tablets project into the Maven artifact.
openl:deploy	Deploys the OpenL Tablets project to an OpenL Tablets repository.
openl:help	Displays help information on openl-maven-plugin.

2.2 Usage Overview

General instructions on how to use the plugin name can be found in [Usage](#). This section provides specific configuration example.

Specify the version in the project plugin configuration as follows:

```
<project>
...
<build>
  <!-- To define the plugin version in your parent POM -->
  <pluginManagement>
    <plugins>
      <plugin>
        <groupId>org.openl.rules</groupId>
        <artifactId>openl-maven-plugin</artifactId>
        <version>${org.openl.version}</version>
      </plugin>
    </plugins>
  </build>
...
```

```
    </plugins>
  </pluginManagement>
  <!-- To use the plugin goals in your POM or parent POM -->
  <plugins>
    <plugin>
      <groupId>org.openl.rules</groupId>
      <artifactId>openl-maven-plugin</artifactId>
      <version>${openl.version}</version>
      <extensions>true</extensions>
    </plugin>
    ...
  </plugins>
</build>
...
</project>
```

3 Goals

This section includes the following topics:

- [openl:generate](#)
- [openl:compile](#)
- [openl:test](#)
- [openl:package](#)
- [openl:deploy](#)
- [openl:help](#)

3.1 openl:generate

Full name:

`org.openl.rules:openl-maven-plugin:5.21.0:generate`

Description:

Generates OpenL Tablets interface, domain classes, project descriptor, and unit tests.

Attributes:

- Requires a Maven project to be executed.
- Requires dependency resolution of artifacts in scope: `compile`.
- By default, binds to the **generate-sources** lifecycle phase.

Required Parameters

Required parameters for openl:generate			
Name	Type	Since	Description
workspaceFolder	File	-	Directory containing the generated artifact. Default value: <code>\${project.build.directory}/openl-workspace</code>

Optional Parameters

Optional parameters for openl:generate			
Name	Type	Since	Description
classpaths	String[]	-	Deprecated. There is no need to generate <code>rules.xml</code> from Maven.
createProjectDescriptor	boolean	-	Deprecated. There is no need to generate <code>rules.xml</code> from Maven. Default value: <code>true</code> .
externalParameters	map	-	Additional options for compilation defined externally, such as external dependencies and overridden system properties.
generateInterfaces	GenerateInterface[]	-	Deprecated. It is replaced with the smart generator. Use <code>interfaceClass</code> instead.

Optional parameters for openl:generate			
Name	Type	Since	Description
generateUnitTests	Boolean	-	Deprecated. Use the openl:test goal to run OpenL Tablets tests. Default value: <code>false</code> .
interfaceClass	String	5.19.1	Generated Java interface from an OpenL Tablets project. If it is empty, generation is skipped.
isProvideRuntimeContext	boolean	5.19.1	Parameter that adds the IRulesRuntimeContext arguments to the generated interface.
isProvideVariations	boolean	5.19.1	Parameter to add more methods to the generated interface to support variations.
openlResourcesDirectory	String	-	Deprecated. Use sourceDirectory instead.
OutputDirectory	String	-	Output directory of generated Java beans and OpenL Tablets Java interface. Default value: <code>\${project.build.directory}/generated-sources/openl</code> .
overwriteProjectDescriptor	Boolean	-	Deprecated. There is no need to generate <code>rules.xml</code> from Maven. Default value: <code>true</code> .
overwriteUnitTests	Boolean	-	Deprecated. Use the openl:test goal to run OpenL Tablets tests. Default value: <code>false</code> .
projectName	String	-	Deprecated. There is no need to generate <code>rules.xml</code> from Maven.
sourceDirectory	File	5.19.0	Folder that contains all OpenL Tablets related resources, such as rules and project descriptor. Example: <code>\${project.basedir}/src/main/openl</code> Default value: <code>\${project.build.sourceDirectory}/../openl</code>
superInterface	String	5.19.1	Comma-separated list of interfaces used to extend the generated interface.
unitTestTemplatePath	String	-	Deprecated. Use the openl:test goal to run OpenL Tablets tests. Default value: <code>org/openl/rules/maven/JUnitTestTemplate.vm</code> .

Parameter Details

classpath:

Deprecated. There is no need to generate `rules.xml` from Maven.

Default classpath entries in `rules.xml`. The default value is `{".", ""}`. It is used only if `createProjectDescriptor == true`.

- **Type:** `java.lang.String[]`
- **Required:** No

createProjectDescriptor:

Deprecated. There is no need to generate `rules.xml` from Maven.

Parameter which denotes, if set to `true`, that `rules.xml` will be generated if it does not exist. The default value is `true`.

- **Type:** `boolean`
- **Required:** No
- **Default:** `true`

externalParameters:

Additional options for compilation defined externally, such as external dependencies and overridden system properties.

- **Type:** `java.util.Map`
- **Required:** No

generateInterfaces:

Deprecated. It is replaced with smart generator. Use `interfaceClass` instead.

Tasks that generate classes or data types.

Object Properties

Object properties			
Name	Type	Required	Description
srcFile	String	false	Reference to the Excel file for which an interface class must be generated.
targetClass	String	false	Full name of the interface class to be generated. It is optional if a missed interface is not generated. OpenL Tablets WebStudio recognizes modules in projects by interface classes and uses their names in the user interface. If there are multiple wrappers with identical names, only one of them is recognized as a module in OpenL Tablets WebStudio.
isUsedRuleXmlForGenerate	boolean	false	Parameter to enable the system generate class and datatypes from rules.xml. If set to yes , <code>srcFile</code> is ignored. <code>targetClass</code> is required for this parameter.
displayName	String	false	End user-oriented title of the file that appears in OpenL Tablets WebStudio. Default value: the Excel file name without extension.
targetSrcDir	String	false	Folder where the generated interface class must be placed. An example is <code>src/main/java</code> . Default value: <code>\${project.build.sourceDirectory}</code> .
openlName	String	false	OpenL configuration to be used. For OpenL Tablets, the <code>org.openl.xls</code> value must always be used. Default value: <code>org.openl.xls</code> .
userHome	String	false	Location of user-defined resources relative to the current OpenL Tablets project. Default value: . The dot stands for the current folder.
userClassPath	String	false	Reference to the folder with additional compiled classes imported by the module when the interface is generated. Default value: <code>null</code> .

Object properties			
Name	Type	Required	Description
ignoreTestMethods	boolean	false	Parameter which denotes, if set to <code>true</code> , that test methods will not be added to interface class. It is used only in <code>GenerateInterface</code> . Default value: <code>true</code> .
generateUnitTests	boolean	false	Parameter that overwrites the base <code>generateUnitTests</code> value.
unitTestTemplatePath	String	false	Parameter that overwrites the base <code>unitTestTemplatePath</code> value.
overwriteUnitTests	boolean	false	Parameter that overwrites the base <code>overwriteUnitTests</code> value.
generateDataType	boolean	false	Parameter that denotes whether <code>dataType</code> must be generated for the current task.

- **Type:** `org.openl.rules.Maven.gen.GenerateInterface[]`
- **Required:** No

generateUnitTests:

Deprecated. Use the `openl:test` goal to run OpenL Tablets tests.

If set to `true`, JUnit tests for OpenL Tablets Test tables will be generated. The default value is `false`.

- **Type:** `java.lang.Boolean`
- **Required:** No
- **Default:** `false`

interfaceClass:

Generated Java interface from an OpenL Tablets project. If it is empty, generation is skipped.

- **Type:** `java.lang.String`
- **Since:** 5.19.1
- **Required:** No

isProvideRuntimeContext:

Parameter that adds `IRulesRuntimeContext` arguments to the generated interface.

- **Type:** `boolean`
- **Since:** 5.19.1
- **Required:** No

isProvideVariations:

Parameter to add additional methods to the generated interface to support variations.

- **Type:** `boolean`
- **Since:** 5.19.1
- **Required:** No

openlResourcesDirectory:

Deprecated. Use `sourceDirectory` instead.

- **Type:** `java.lang.String`
- **Required:** No

OutputDirectory:

Output directory of generated Java beans and OpenL Tablets Java interface.

Type: `java.io.File`

Required: No

Default: `${project.build.directory}/generated-sources/openl`

overwriteProjectDescriptor:

Deprecated. There is no need to generate `rules.xml` from Maven.

If set to `true`, `rules.xml` will be overwritten on each run. If set to `false`, `rules.xml` generation will be skipped if it exists. Makes sense only if `createProjectDescriptor == true`. The default value is `"true"`.

- **Type:** `boolean`
- **Required:** No
- **Default:** `true`

overwriteUnitTests:

Deprecated. Use the `openl:test` goal to run OpenL Tablets tests.

If set to `true`, existing JUnit tests will be overwritten. If set to `false`, only absent tests will be generated, and others will be skipped.

- **Type:** `java.lang.Boolean`
- **Required:** No
- **Default:** `false`

projectName:

Deprecated. There is no need to generate `rules.xml` from Maven.

Default project name in `rules.xml`. If omitted, the name of the first module in the project is used. The parameter is used only if `createProjectDescriptor == true`.

- **Type:** `java.lang.String`
- **Required:** No

sourceDirectory:

Folder that contains all OpenL Tablets related resources, such as rules and project descriptor.

Example: `${project.basedir}/src/main/openl`

- **Type:** `java.io.File`
- **Since:** 5.19.0
- **Required:** No
- **Default:** `${project.build.sourceDirectory}/../openl`

superInterface:

Comma-separated list of interfaces used to extend the generated interface.

- **Type:** `java.lang.String`
- **Since:** 5.19.1
- **Required:** No

unitTestTemplatePath:

Deprecated. Use the `openl:test` goal to run OpenL Tablets tests.

Path to Velocity template for generated unit tests. If omitted, a default template will be used. The parameter is available in template variables:

Variables	
Name	Description
<code>openlInterfacePackage</code>	Package of generated interface class.
<code>openlInterfaceClass</code>	Generated interface class name.
<code>testMethodNames</code>	Available test method names.
<code>projectRoot</code>	Root directory of the OpenL Tablets project.
<code>srcFile</code>	Reference to the Excel file for which an interface class must be generated.
<code>StringUtils</code>	Apache commons utility class.

- **Type:** `java.lang.String`
- **Required:** No
- **Default:** `org/openl/rules/maven/JUnitTestTemplate.vm`

workspaceFolder:

Directory containing the generated artifact.

- **Type:** `java.io.File`
- **Required:** Yes
- **Default:** `${project.build.directory}/openl-workspace`

3.2 openl:compile

Full name:

`org.openl.rules:openl-maven-plugin:5.21.0:compile`

Description:

Compiles and validates OpenL Tablets project.

Attributes:

- Requires a Maven project to be executed.
- Requires dependency resolution of artifacts in scope: **compile**.
- Binds by default to the **compile** lifecycle phase.

Required Parameters

Required parameters for openl:compile			
Name	Type	Since	Description
<code>workspaceFolder</code>	File	-	Directory containing the generated artifact.
Default value:			
<code>\${project.build.directory}/openl-workspace</code>			

Optional Parameters

Optional parameters for openl:compile			
Name	Type	Since	Description
externalParameters	Map	-	Additional options for compilation defined externally, such as external dependencies and overridden system properties.
openlResourcesDirectory	String	-	Deprecated. Use sourceDirectory instead.
sourceDirectory	File	5.19.0	Folder that contains all OpenL Tablets related resources, such as rules and project descriptor. Example: <code>\${project.basedir}/src/main/openl</code> Default value: <code>\${project.build.sourceDirectory}/../openl</code>

Parameter Details

externalParameters:

Additional options for compilation defined externally, such as external dependencies and overridden system properties.

- **Type:** `java.util.Map`
- **Required:** No

openlResourcesDirectory:

Deprecated. Use sourceDirectory instead.

- **Type:** `java.lang.String`
- **Required:** No

sourceDirectory:

Folder that contains all OpenL Tablets related resources, such as rules and project descriptor.

Example: `${project.basedir}/src/main/openl`

- **Type:** `java.io.File`
- **Since:** 5.19.0
- **Required:** No
- **Default:** `${project.build.sourceDirectory}/../openl`

workspaceFolder:

Directory containing the generated artifact.

- **Type:** `java.io.File`
- **Required:** Yes
- **Default:** `${project.build.directory}/openl-workspace`

3.3 openl:test

Full name:

`org.openl.rules:openl-maven-plugin:5.21.0:test`

Description:

Runs OpenL Tablets tests.

Attributes:

- Requires a Maven project to be executed.
- Requires dependency resolution of artifacts in scope: test.
- Binds by default to the **test** lifecycle phase.

Required Parameters

Required parameters for openl:test			
Name	Type	Since	Description
workspaceFolder	File	-	Directory containing the generated artifact. Default value: <code>\${project.build.directory}/openl-workspace</code>

Optional Parameters

Optional parameters for openl:test									
Name	Type	Since	Description						
externalParameters	Map	-	Additional options for compilation defined externally, such as external dependencies and overridden system properties.						
openlResourcesDirectory	String	-	Deprecated. Use sourceDirectory instead.						
singleModuleMode	boolean	-	Parameter for compiling the project in the single module mode. This parameter is required for big projects. If it is set to <code>true</code> , each module is compiled in sequence and test from that module are run. If it is set to <code>false</code> , all modules are compiled at once and all tests from all modules are run. The default value is <code>false</code> .						
skipTests	boolean		Parameter which denotes, if set to <code>true</code> , to skip running OpenL Tablets tests. User property: <code>skipTests</code> .						
sourceDirectory	File	5.19.0	Folder that contains all OpenL Tablets related resources, such as rules and project descriptor. Example: <code>\${project.basedir}/src/main/openl</code> Default value: <code>\${project.build.sourceDirectory}/../openl</code>						
testSourceDirectory	File	-	Directory containing OpenL Tablets sources to be used in testing OpenL Tablets rules. Default value: <code>\${project.build.testSourceDirectory}/../openl</code>						
threadCount	String		Thread count to run test cases. The values are as follows: <table><tr><th>Value</th><th>Description</th></tr><tr><td>4</td><td>Runs tests with 4 threads.</td></tr><tr><td>1.5C</td><td>Runs tests with 1.5 threads per CPU core.</td></tr></table>	Value	Description	4	Runs tests with 4 threads.	1.5C	Runs tests with 1.5 threads per CPU core.
Value	Description								
4	Runs tests with 4 threads.								
1.5C	Runs tests with 1.5 threads per CPU core.								

Optional parameters for <code>openl:test</code>			
Name	Type	Since	Description
		none	Runs tests sequentially. No threads are created to run tests.
		auto	Automatically configures thread count.
Default value: auto			

Parameter Details

`externalParameters:`

Additional options for compilation defined externally, such as external dependencies and overridden system properties.

- **Type:** `java.util.Map`
- **Required:** No

`openlResourcesDirectory:`

Deprecated. Use `sourceDirectory` instead.

- **Type:** `java.lang.String`
- **Required:** No

`singleModuleMode:`

Parameter for compiling the project in the single module mode. This parameter is required for big projects. If it is set to `true`, each module is compiled in sequence and test from that module are run. If it is set to `false`, all modules are compiled at once and all tests from all modules are run.

The default value is `false`.

- **Type:** `boolean`
- **Required:** No
- **Default:** `false`

`skipTests:`

Parameter which denotes, if set to `true`, to skip running OpenL Tablets tests.

- **Type:** `boolean`
- **Required:** No
- **User property:** `skipTests`

`sourceDirectory:`

Folder that contains all OpenL Tablets related resources, such as rules and project descriptor.

Example: `${project.basedir}/src/main/openl`

- **Type:** `java.io.File`
- **Since:** 5.19.0
- **Required:** No
- **Default:** `${project.build.sourceDirectory}/../openl`

`testSourceDirectory:`

Directory containing OpenL Tablets sources to be used in testing OpenL Tablets rules.

- **Type:** `java.io.File`
- **Required:** No
- **Default:** `${project.build.testSourceDirectory}/../openl`

threadCount:

Thread count to run test cases. The values are as follows

threadCount parameter values	
Value	Description
4	Runs tests with 4 threads.
1.5C	Runs tests with 1.5 threads per CPU core.
none	Runs tests sequentially. No threads are created to run tests.
auto	Automatically configures thread count.

The default value is `auto`.

- **Type:** `java.lang.String`
- **Required:** No
- **Default:** `auto`

workspaceFolder:

Directory containing the generated artifact.

- **Type:** `java.io.File`
- **Required:** Yes
- **Default:** `${project.build.directory}/openl-workspace`

3.4 openl:package

Full name:

`org.openl.rules:openl-maven-plugin:5.21.0:package`

Description:

Packages an OpenL Tablets project in a ZIP archive.

Attributes:

- Requires a Maven project to be executed.
- Requires dependency resolution of artifacts in scope: `compile+runtime`.
- Since version: 5.19.1.
- Binds by default to the **package** lifecycle phase.

Required Parameters

Required parameters for openl:package			
Name	Type	Since	Description
dependenciesThreshold	int	5.19.1	Allowed quantity of dependencies which can be included into the ZIP archive. Usually OpenL Tablets rules require a few dependencies, such as domain models, that is, Java beans, or some utils, for example, JSON parsing. Usually the quantity of required dependencies does not exceed 3. If transitive dependencies are declared incorrectly, the size of the ZIP package increases dramatically. This parameter allows preventing such situation by failing packaging. Default value is: 3.
outputDirectory	File	5.19.1	Directory containing the generated artifact. Default value: <code>\${project.build.directory}</code>
workspaceFolder	File	5.19.1	Directory containing the generated artifact. Default value: <code>\${project.build.directory}/openl-workspace</code>

Optional Parameters

Optional parameters for openl:package			
Name	Type	Since	Description
classifier	String	5.19.1	Parameter that packages an OpenL Tablets project as a supplemental artifact with the given classifier. By default, the classifier is not provided, and only the OpenL Tablets project is created as a main artifact. Upon the attempt to create the second main artifact, without a classifier, the build fails.
classpathFolder	String	5.19.1	Folder to store dependencies inside the OpenL Tablets project. Default value: <code>lib/</code> .
format	String	5.19.1	Comma separated list of packaging formats. Default value: <code>zip</code> .
openlResourcesDirectory	String	5.19.1	Deprecated. Use <code>sourceDirectory</code> instead.
sourceDirectory	File	5.19.0	Folder that contains all OpenL Tablets related resources, such as rules and project descriptor. Example: <code>\${project.basedir}/src/main/openl</code> Default value: <code>\${project.build.sourceDirectory}/../openl</code>

Parameter Details

classifier:

Parameter that packages an OpenL Tablets project as a supplemental artifact with the given classifier. By default, the classifier is not provided, and only the OpenL Tablets project is created as a main artifact. Upon the attempt to create the second main artifact, without a classifier, the build fails.

- **Type:** `java.lang.String`
- **Since:** 5.19.1
- **Required:** No

classpathFolder:

Folder to store dependencies inside the OpenL Tablets project.

- **Type:** `java.lang.String`
- **Since:** 5.19.1
- **Required:** No
- **Default:** `lib/`

dependenciesThreshold:

Allowed quantity of dependencies which can be included into the ZIP archive. Usually OpenL Tablets rules require a few dependencies, such as domain models, that is, Java beans, or some utils, for example, JSON parsing. Usually the quantity of required dependencies does not exceed 3. If transitive dependencies are declared incorrectly, the size of the ZIP package increases dramatically. This parameter allows preventing such situation by failing packaging.

- **Type:** `int`
- **Since:** 5.19.1
- **Required:** Yes
- **Default:** 3

format:

Comma separated list of packaging formats.

- **Type:** `java.lang.String`
- **Since:** 5.19.1
- **Required:** No
- **Default:** `zip`

openlResourcesDirectory:

Deprecated. Use `sourceDirectory` instead.

- **Type:** `java.lang.String`
- **Since:** 5.19.1
- **Required:** No

outputDirectory:

Directory containing the generated artifact.

- **Type:** `java.io.File`
- **Since:** 5.19.1
- **Required:** Yes
- **Default:** `${project.build.directory}`

sourceDirectory:

Folder that contains all OpenL Tablets related resources, such as rules and project descriptor.

Example: `${project.basedir}/src/main/openl`

- **Type:** `java.io.File`
- **Since:** 5.19.0
- **Required:** No

- **Default:** `${project.build.sourceDirectory}/../openl`

workspaceFolder:

Directory containing the generated artifact.

- **Type:** `java.io.File`
- **Since:** 5.19.1
- **Required:** Yes
- **Default:** `${project.build.directory}/openl-workspace`

3.5 openl:deploy

Full name:

`org.openl.rules:openl-maven-plugin:5.21.0:deploy`

Description:

Deploys the OpenL Tablets project to an OpenL Tablets repository.

Attributes:

- Requires a Maven project to be executed.
- Requires dependency resolution of artifacts in scope: **compile**.
- Binds by default to the **deploy** lifecycle phase.

Required Parameters

Required parameters for openl:deploy			
Name	Type	Since	Description
deployServer	String	-	User property: <code>openl.deployServer</code>
deployType	String	-	User property: <code>openl.deployType</code>
deployUrl	String	-	User property: <code>openl.deployUrl</code>
outputDirectory	File	-	Directory containing the generated artifact. Default value: <code>\${project.build.directory}</code>
workspaceFolder	File	-	Directory containing the generated artifact. Default value: <code>\${project.build.directory}/openl-workspace</code>

Optional Parameters

Optional parameters for openl:deploy			
Name	Type	Since	Description
openlResourcesDirectory	String	-	Deprecated. Use <code>sourceDirectory</code> instead.
sourceDirectory	File	5.19.0	Folder that contains all OpenL Tablets related resources, such as rules and project descriptor. Example: <code>\${project.basedir}/src/main/openl</code> Default value: <code>\${project.build.sourceDirectory}/../openl</code>

Parameter Details

deployServer:

- **Type:** `java.lang.String`
- **Required:** Yes
- **User Property:** `openl.deployServer`

deployType:

- **Type:** `java.lang.String`
- **Required:** Yes
- **User Property:** `openl.deployType`

deployUrl:

- **Type:** `java.lang.String`
- **Required:** Yes
- **User Property:** `openl.deployUrl`

openlResourcesDirectory:

Deprecated. Use `sourceDirectory` instead.

- **Type:** `java.lang.String`
- **Required:** No

outputDirectory:

Directory containing the generated artifact.

- **Type:** `java.io.File`
- **Required:** Yes
- **Default:** `${project.build.directory}`

sourceDirectory:

Folder that contains all OpenL Tablets related resources, such as rules and project descriptor.

Example: `${project.basedir}/src/main/openl`

- **Type:** `java.io.File`
- **Since:** 5.19.0
- **Required:** No
- **Default:** `${project.build.sourceDirectory}/../openl`

workspaceFolder:

Directory containing the generated artifact.

- **Type:** `java.io.File`
- **Required:** Yes
- **Default:** `${project.build.directory}/openl-workspace`

3.6 openl:help

Full name:

org.openl.rules:openl-maven-plugin:5.21.0:help

Description:

Displays help information on openl-maven-plugin.

Calls `mvn openl:help -Ddetail=true -Dgoal=<goal-name>` to display parameter details.

Optional Parameters

Optional parameters for openl:help			
Name	Type	Since	Description
detail	boolean	-	Parameter which denotes, if set to <code>true</code> , to display all settable properties for each goal. Default value: <code>false</code> . User property: <code>detail</code> .
goal	String	-	Name of the goal for which to display help. If left unspecified, all goals are displayed. User property: <code>goal</code> .
indentSize	int	-	Number of spaces per indentation level. The value must be positive. Default value: <code>2</code> . User property: <code>indentSize</code> .
lineLength	int	-	Maximum length of a display line. The value must be positive. Default value: <code>80</code> . User property: <code>lineLength</code> .

Parameter Details

detail:

Parameter which denotes, if set to `true`, to display all settable properties for each goal.

- **Type:** `boolean`
- **Required:** No
- **User Property:** `detail`
- **Default:** `false`

goal:

Name of the goal for which to display help. If left unspecified, all goals are displayed.

- **Type:** `java.lang.String`
- **Required:** No
- **User Property:** `goal`

indentSize:

Number of spaces per indentation level. The value must be positive.

- **Type:** `int`
- **Required:** No

- **User Property:** `indentSize`
- **Default:** `2`

lineLength:

Maximum length of a display line. The value must be positive.

- **Type:** `int`
- **Required:** No
- **User Property:** `lineLength`
- **Default:** `80`

4 Usage

This section includes the following topics:

- [Directory Structure](#)
- [Configure Interface, Domain Classes and Project Descriptor Generation](#)
- [Configure OpenL Project Compilation and Validation](#)
- [Configure OpenL Project Testing](#)

4.1 Directory Structure

OpenL Maven Plugin expects the following directory structure:

```

|- your-project/           Project root folder
| |- pom.xml              Maven project file
| |
| |- src/
| |
| | |- main/
| | |
| | | |- java/            Contains Java sources
| | | |
| | | |- resources/       Contains Java resources
| | | |
| | | |- openl/           Contains all OpenL Tablets-related resources (rules, xml)
| | | |
| | | | |- rules.xml      OpenL Tablets project descriptor (for OpenL Tablets only)
| | | | |- rules/
| | | | |- TemplateRules.xls  File with rules

```

Note that OpenL Tablets-related resources are located in the `src/main/openl` directory. It can be changed to fit user needs by modifying the `sourceDirectory` parameter in Maven plugin configuration.

Note: It is not recommended to put OpenL Tablets-related resources to the `src/main/resources` folder. In this case, OpenL Tablets resources will be inside the JAR file alongside with the compiled Java classes, which most probably is not what was expected to do in production.

4.2 Integrate openl-maven-plugin in the Maven Lifecycle

The simplest way to create an OpenL Tablets artefact is as follows:

```

<packaging>openl</packaging>
[...]
<build>
  [...]
  <plugins>
    [...]
    <plugin>
      <groupId>org.openl.rules</groupId>
      <artifactId>openl-maven-plugin</artifactId>
      <version>${org.openl.version}</version>
      <extensions>true</extensions>
    </plugin>
  </plugins>
</build>

```

```

        </plugin>

    </plugins>
    [...]
</build>

```

In this case, the OpenL Tablets project is processed through the Maven lifecycle as follows:

- Java compilation happens at the **compile** phase.
- OpenL Tablets rules are validated at the **compile** phase.
- OpenL Tablets tests are run at the **test** phase.
- ZIP archive is created and registered as a Maven artifact at the **package** phase.

4.3 Configure Interface, Domain Classes and Project Descriptor Generation

The simplest way to generate interface for rules is defined in the `TemplateRules.xls` file as follows:

```

<build>
  [...]
  <plugins>
    [...]
    <plugin>
      <groupId>org.openl.rules</groupId>
      <artifactId>openl-maven-plugin</artifactId>
      <version>${org.openl.version}</version>
      <configuration>
        <interfaceClass>org.company.gen.TemplateRulesInterface</interfaceClass>
      </configuration>
      <executions>
        <execution>
          <goals>
            <goal>generate</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
  [...]
</build>

```

In this case, classes are generated on each Maven run during the **generate-sources** phase.

To invoke class generation manually, remove the `executions` node and run in the console when needed:

```
mvn openl:generate
```

For more information on configuration options, see [openl:generate](#).

4.4 Configure OpenL Project Compilation and Validation

```

<build>
  [...]
  <plugins>
    [...]
    <plugin>

```



```
<groupId>org.openl.rules</groupId>
<artifactId>openl-maven-plugin</artifactId>
<version>${org.openl.version}</version>
<executions>
  <execution>
    <goals>
      <goal>compile</goal>
    </goals>
  </execution>
</executions>
</plugin>

</plugins>
[...]
```

For more information on configuration options, see [openl:compile](#).

4.5 Configure OpenL Project Testing

The simplest way to invoke the OpenL Tablets test is as follows:

```
<build>
[...]
```

```
<plugins>
[...]
```

```
<plugin>
  <groupId>org.openl.rules</groupId>
  <artifactId>openl-maven-plugin</artifactId>
  <version>${org.openl.version}</version>
  <executions>
    <execution>
      <goals>
        <goal>test</goal>
      </goals>
    </execution>
  </executions>
</plugin>

</plugins>
[...]
```

For more information on configuration options, see [openl:test](#).

5 Examples

To acquire a better understanding of Plugin Name usage, see the following examples:

- [Configuration with all OpenL Maven Plugin Goals](#)
- [Creating a Project with a Working Example of OpenL Maven Plugin Usage](#)

5.1 Configuration with all OpenL Maven Plugin Goals

To configure `rules.xml` generation, set its project ID, project name, and classpath values. An example is as follows:

```
<build>
  [...]
  <plugins>
    [...]
    <plugin>
      <groupId>org.openl.rules</groupId>
      <artifactId>openl-maven-plugin</artifactId>
      <version>${openl.rules.version}</version>
      <configuration>
        <!-- generate goal -->
        <interfaceClass>com.example.Service</interfaceClass>
        <superInterface>com.example.Base, com.example.Custom</superInterface>
        <isProvideRuntimeContext>true</isProvideRuntimeContext>
        <isProvideVariations>true</isProvideVariations>
        <!-- test goal -->
        <skipTests>false</skipTests>
        <threadCount>auto</threadCount>
        <!-- package -->
        <format>zip,jar</format>
        <classifier>openl</classifier>
        <classpathFolder>lib/</classpathFolder>
        <!-- common part -->
        <sourceDirectory>src/main/openl</sourceDirectory>
      </configuration>
      <executions>
        <execution>
          <goals>
            <goal>generate</goal>
            <goal>compile</goal>
            <goal>test</goal>
            <goal>package</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
  [...]
</build>
```

For more information on configuration options, see [openl:generate](#).

5.2 Creating a Project with a Working Example of OpenL Maven Plugin Usage

OpenL Tablets has an archetype which can be used to create a simple OpenL Rules project containing an example of OpenL Maven plugin usage. Proceed as follows:

1. Execute the following command in the command line:

```
mvn archetype:generate
```

Maven runs an archetype console wizard.

2. Select the `openl-simple-project-archetype` menu item.
3. Follow the wizard instructions to complete project creation.

When the creation is completed, a new Maven-based project appears in the file system. It is an OpenL Tablets Rules project which has one module with simple rules.

4. To compile the project, in the command line, execute the following command from the root of the project folder:

```
mvn install
```

After that, the following objects can be found in the `target` folder:

- A ZIP file without a suffix for importing a project to OpenL Tablets WebStudio.
For more information, see ***[EIS_Suite_OpenL_Tablets_WebStudio_(UG)]***.
- A ZIP file with "-runnable" suffix that can be executed after unpacking.
It demonstrates how OpenL Tablets rules can be invoked from the Java code.
- A JAR file that contains only compiled Java classes.