

maven properties

7.1 Variables in Maven – properties

Frequently encountered parameters Maven allows you to put into variables. This is very useful when you need to match the parameters in different parts of the pom file. For example, you can put the Java version, library versions, paths to certain resources into a variable.

For this, there is a special section in pom.xml – `<properties>`, in which variables are declared. The general form of the variable is as follows:

```
<variable-name> _ _ _ _meaning< / variable name > _
```

Example:

```
<properties>
  <junit.version>5.2</junit.version>
  <project.artifactId>new-app</project.artifactId>
  <maven.compiler.source>1.13</maven.compiler.source>
  <maven.compiler.target>1.15</maven.compiler.target>
</properties>
```

Variables are accessed using a different syntax:

```
$ { variable -name } _
```

Where such code is written, Maven will substitute the value of the variable.

Example:

```

<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>${junit.version}</version>
    <scope>test</scope>
  </dependency>
</dependencies>

<build>
  <finalName>${project.artifactId}</finalName>
  <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>2.3.2</version>
    <configuration>
      <source>${maven.compiler.source}</source>
      <target>${maven.compiler.target}</target>
    </configuration>
  </plugin>
</build>

```

7.2 Predefined variables in Maven

When describing a project in a pom file, you can use predefined variables. They can be conditionally divided into several groups:

- Built-in project properties;
- Project properties;
- Settings.

There are only two built-in project properties:

Property	Description
<code>\${basedir}</code>	project root directory where <code>pom.xml</code>
<code>\${version}</code>	artifact version; can be used <code>\${project.version}</code> or <code>\${pom.version}</code>

Project properties can be referenced using the «project» or «pom» prefixes. We have four of them:

Property	Description
<code>\${project.build.directory}</code>	<code>«target»</code> project directory
<code>\${project.build.outputDirectory}</code>	<code>«target»</code> compiler directory. Default <code>«target/classes»</code>
<code>\${project.name}</code>	name of the project
<code>\${project.version}</code>	project version

Properties `settings.xml` can be accessed using the prefix `settings`. Names can be anything - they are taken from `settings.xml`. Example:

```
${settings.localRepository} sets the path to the local repository.
```