

## Usage

The **Javadoc Plugin** generates **javadocs** using the **Javadoc tool**. The following examples describe the basic usage of the Plugin.

### ➔ Generate Javadocs As Part Of Project Reports

To generate **javadocs** as part of the **site generation**, you should add the Javadoc Plugin in the **<reporting>** section of your **pom**:

```
<project>
...
<reporting>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-javadoc-plugin</artifactId>
      <version>2.10.3</version>
      <configuration>
        ...
      </configuration>
    </plugin>
  </plugins>
  ...
</reporting>
...
</project>
```

When you execute **mvn site**, the **javadocs** will be generated and included in the generated **site**. A link to the javadocs will be added in the **Project Reports menu**.

### ➔ Generate Standalone Javadocs

To generate **standalone javadocs** for the project, you could add the Javadoc Plugin in the **<build>** section of your **pom** (if no configuration defined, the plugin uses default values):

```
<project>
...
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-javadoc-plugin</artifactId>
      <version>2.10.3</version>
      <configuration>
        ...
      </configuration>
    </plugin>
  </plugins>
  ...
</build>
...
</project>
```

And execute any of the following commands:

```
mvn javadoc:javadoc
mvn javadoc:jar
mvn javadoc:aggregate
```


```
mvn javadoc:aggregate-jar
mvn javadoc:test-javadoc
mvn javadoc:test-jar
mvn javadoc:test-aggregate
mvn javadoc:test-aggregate-jar
```

For all `jar` goals, the javadocs are first generated and then packaged into a `jar` file.

## ➔ Javadoc Configuration

The Javadoc Plugin supports a large number of configuration parameters. Each configuration parameter turns into a tag name.

Please refer to the [Javadoc Plugin Documentation](#) for a listing of these parameters. Most of these parameters are passed directly to the Javadoc tool itself.

**IMPORTANT NOTE:** configuring the Javadoc plugin in the `<reporting/>` or `<build/>` elements in the pom have **NOT** the same behavior as described in the [Guide to Configuring Plug-ins](#) .

For instance, if you have the following snippet:

```
<project>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-javadoc-plugin</artifactId>
        <version>2.10.3</version>
        <configuration>
          <show>private</show>
          <nohelp>true</nohelp>
        </configuration>
      </plugin>
    </plugins>
  </build>

  <reporting>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-javadoc-plugin</artifactId>
        <version>2.10.3</version>
        <configuration>
          <stylesheetfile>${basedir}/src/main/javadoc/stylesheet.css</stylesheetfile>
          <show>public</show>
        </configuration>
      </plugin>
    </plugins>
  </reporting>
</build>
```

## ➔ mvn site

It will generate the Javadoc for public members (defined in `<reporting/>`) using the given stylesheet (defined in `<reporting/>`), and with an help page (default value for `nohelp` is true).

## ➔ mvn javadoc:javadoc

It will generate the Javadoc for private members (defined in `<build/>`) using the stylesheet (defined in `<reporting/>`), and with no help page (defined in `<build/>`).