

20+ Maven Commands and Options (Cheat Sheet)

 journaldev.com/33645/maven-commands-options-cheat-sheet

Maven Commands

Let's look into some popular and must know maven commands. We will use a sample Maven project to showcase the command output.

1. mvn clean

This command cleans the maven project by deleting the target directory. The command output relevant messages are shown below.

```
$ mvn clean
...
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ maven-example-jar ---
[INFO] Deleting /Users/pankaj/Desktop/maven-examples/maven-example-jar/target
...
$
```

2. mvn compiler:compile

This command compiles the java source classes of the maven project.

```
$ mvn compiler:compile
...
[INFO] --- maven-compiler-plugin:3.8.1:compile (default-cli) @ maven-example-jar -
--
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /Users/pankaj/Desktop/maven-examples/maven-
example-jar/target/classes
...
$
```

3. mvn compiler:testCompile

This command compiles the test classes of the maven project.

```
$ mvn compiler:testCompile
...
[INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-cli) @ maven-example-
jar ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /Users/pankaj/Desktop/maven-examples/maven-
example-jar/target/test-classes
...
$
```

4. mvn package

This command builds the maven project and packages them into a JAR, WAR, etc.

```

$ mvn package
...
[INFO] --- maven-compiler-plugin:3.8.1:compile (default-compile) @ maven-example-jar ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /Users/pankaj/Desktop/maven-examples/maven-example-jar/target/classes
...
[INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ maven-example-jar ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /Users/pankaj/Desktop/maven-examples/maven-example-jar/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ maven-example-jar ---
-
[INFO] Surefire report directory: /Users/pankaj/Desktop/maven-examples/maven-example-jar/target/surefire-reports

-----
T E S T S
-----
Running com.journaldev.maven.classes.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.005 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ maven-example-jar ---
[INFO] Building jar: /Users/pankaj/Desktop/maven-examples/maven-example-jar/target/maven-example-jar-0.0.1-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
...

```

The output shows the location of the JAR file just before the “BUILD SUCCESS” message. Notice the package goal executes compile, testCompile, and test goals before packaging the build.

5. mvn install

This command builds the maven project and installs the project files (JAR, WAR, pom.xml, etc) to the local repository.

```

$ mvn install
...
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ maven-
example-jar ---
...
[INFO] --- maven-compiler-plugin:3.8.1:compile (default-compile) @ maven-example-
jar ---
...
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @
maven-example-jar ---
...
[INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ maven-
example-jar ---
...
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ maven-example-jar --
-
...
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ maven-example-jar ---
...
[INFO] --- maven-install-plugin:2.4:install (default-install) @ maven-example-jar
---
[INFO] Installing /Users/pankaj/Desktop/maven-examples/maven-example-jar-
target/maven-example-jar-0.0.1-SNAPSHOT.jar to
/Users/pankaj/.m2/repository/com/journaldev/maven/maven-example-jar/0.0.1-
SNAPSHOT/maven-example-jar-0.0.1-SNAPSHOT.jar
[INFO] Installing /Users/pankaj/Desktop/maven-examples/maven-example-jar/pom.xml
to /Users/pankaj/.m2/repository/com/journaldev/maven/maven-example-jar/0.0.1-
SNAPSHOT/maven-example-jar-0.0.1-SNAPSHOT.pom
...

```

6. mvn deploy

This command is used to deploy the artifact to the remote repository. The remote repository should be configured properly in the project pom.xml file distributionManagement tag. The server entries in the maven settings.xml file is used to provide authentication details.

7. mvn validate

This command validates the maven project that everything is correct and all the necessary information is available.

8. mvn dependency:tree

This command generates the dependency tree of the maven project.

```
$ mvn dependency:tree
...
[INFO] --- maven-dependency-plugin:2.8:tree (default-cli) @ Mockito-Examples ---
[INFO] com.journaldev.mockito:Mockito-Examples:jar:1.0-SNAPSHOT
[INFO] +- org.junit.platform:junit-platform-runner:jar:1.2.0:test
[INFO] | +- org.apiguardian:apiguardian-api:jar:1.0.0:test
[INFO] | +- org.junit.platform:junit-platform-launcher:jar:1.2.0:test
[INFO] | \- org.junit.platform:junit-platform-suite-api:jar:1.2.0:test
[INFO] |     \- org.junit.platform:junit-platform-commons:jar:1.2.0:test
[INFO] +- org.junit.jupiter:junit-jupiter-engine:jar:5.2.0:test
[INFO] | +- org.junit.platform:junit-platform-engine:jar:1.2.0:test
[INFO] | | \- org.opentest4j:opentest4j:jar:1.1.0:test
[INFO] | \- org.junit.jupiter:junit-jupiter-api:jar:5.2.0:test
[INFO] +- org.mockito:mockito-junit-jupiter:jar:2.19.0:test
[INFO] | \- org.mockito:mockito-core:jar:2.19.0:test
[INFO] |     +- net.bytebuddy:byte-buddy:jar:1.8.10:test
[INFO] |     +- net.bytebuddy:byte-buddy-agent:jar:1.8.10:test
[INFO] |     \- org.objenesis:objenesis:jar:2.6:test
[INFO] \- org.testng:testng:jar:6.14.3:test
[INFO]     +- com.beust:jcommander:jar:1.72:test
[INFO]     \- org.apache-extras.beanshell:bsh:jar:2.0b6:test
```

9. mvn dependency:analyze

This command analyzes the maven project to identify the unused declared and used undeclared dependencies. It's useful in reducing the build size by identifying the unused dependencies and then remove it from the pom.xml file.

```
$ mvn dependency:analyze
...
[INFO] --- maven-dependency-plugin:2.8:analyze (default-cli) @ Mockito-Examples --
-
[WARNING] Used undeclared dependencies found:
[WARNING]   org.junit.jupiter:junit-jupiter-api:jar:5.2.0:test
[WARNING]   org.mockito:mockito-core:jar:2.19.0:test
[WARNING] Unused declared dependencies found:
[WARNING]   org.junit.platform:junit-platform-runner:jar:1.2.0:test
[WARNING]   org.junit.jupiter:junit-jupiter-engine:jar:5.2.0:test
[WARNING]   org.mockito:mockito-junit-jupiter:jar:2.19.0:test
...
$
```

10. mvn archetype:generate

Maven archetypes is a maven project templating toolkit. We can use this command to generate a skeleton maven project of different types, such as JAR, web application, maven site, etc.

Recommended Reading: [Creating a Java Project using Maven Archetypes](#)

11. mvn site:site

This command generates a site for the project. You will notice a “site” directory in the target after executing this command. There will be multiple HTML files inside the site directory that provides information related to the project.

Mockito-Examples

Last Published: 2019-12-14 | Version: 1.0-SNAPSHOT

Project Documentation

- ▼ Project Information
 - Dependencies
 - Dependency
 - Convergence
 - Dependency Information
- About**
- Plugin Management
- Plugins
- Summary



About Mockito-Examples

There is currently no description associated with this project.

Maven site Command

12. mvn test

This command is used to run the test cases of the project using the `maven-surefire-plugin`.

```
$ mvn test
...
[INFO] --- maven-surefire-plugin:2.22.0:test (default-test) @ Mockito-Examples ---
[INFO]
[INFO] -----
[INFO]  T E S T S
[INFO] -----
[INFO] Running TestSuite
first-element
second-element
Employee setName Argument = Pankaj
...
[INFO] Results:
[INFO]
[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0
...
$
```

13. mvn compile

It's used to compile the source Java classes of the project.

```
$ mvn compile
...
[INFO] --- maven-compiler-plugin:3.7.0:compile (default-compile) @ Mockito-Examples ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 10 source files to /Users/pankaj/Desktop/maven-examples/Mockito-Examples/target/classes
...
$
```

14. mvn verify

This command build the project, runs all the test cases and run any checks on the results of the integration tests to ensure quality criteria are met.

Maven Options

Maven provides a lot of command-line options to alter the maven build process. Let's look at some of the important maven options.

15. mvn -help

This command prints the maven usage and all the available options for us to use.

```
→ Mockito-Examples mvn -help

usage: mvn [options] [<goal(s)>] [<phase(s)>]

Options:
  -am,--also-make                If project list is specified, also
                                build projects required by the
                                list
  -amd,--also-make-dependents    If project list is specified, also
                                build projects that depend on
                                projects on the list
  -B,--batch-mode                Run in non-interactive (batch)
                                mode (disables output color)
  -b,--builder <arg>            The id of the build strategy to
                                use
  -C,--strict-checksums          Fail the build if checksums don't
                                match
  -c,--lax-checksums             Warn if checksums don't match
  -cpu,--check-plugin-updates    Ineffective, only kept for
                                backward compatibility
```

Maven Help

16. mvn -f maven-example-jar/pom.xml package

This command is used to build a project from a different location. We are providing the pom.xml file location to build the project. It's useful when you have to run a maven build from a script.

17. mvn -o package

This command is used to run the maven build in the offline mode. It's useful when we have all the required JARs download in the local repository and we don't want Maven to look for any JARs in the remote repository.

18. mvn -q package

Runs the maven build in the quiet mode, only the test cases results and errors are displayed.

```
$ mvn -q package
```

```
-----  
T E S T S  
-----
```

```
Running com.journaldev.maven.classes.AppTest
```

```
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.006 sec
```

```
Results :
```

```
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
```

```
$
```

19. mvn -X package

Prints the maven version and runs the build in the debug mode. It's opposite of the quiet mode and you will see a lot of debug messages in the console.

```
[→ maven-example-jar mvn -X package  
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)  
Maven home: /Users/pankaj/Downloads/apache-maven-3.6.3  
Java version: 13.0.1, vendor: Oracle Corporation, runtime: /Library/Java  
Default locale: en_IN, platform encoding: UTF-8  
OS name: "mac os x", version: "10.15.1", arch: "x86_64", family: "mac"  
[DEBUG] Created new class realm maven.api  
[DEBUG] Importing foreign packages into class realm maven.api  
[DEBUG]   Imported: javax.annotation.* < plexus.core  
[DEBUG]   Imported: javax.annotation.security.* < plexus.core  
[DEBUG]   Imported: javax.enterprise.inject.* < plexus.core  
[DEBUG]   Imported: javax.enterprise.util.* < plexus.core  
[DEBUG]   Imported: javax.inject.* < plexus.core
```

mvn -X Debug Mode

20. mvn -v

Used to display the maven version information.

```
$ mvn -v  
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)  
Maven home: /Users/pankaj/Downloads/apache-maven-3.6.3  
Java version: 13.0.1, vendor: Oracle Corporation, runtime:  
/Library/Java/JavaVirtualMachines/jdk-13.0.1.jdk/Contents/Home  
Default locale: en_IN, platform encoding: UTF-8  
OS name: "mac os x", version: "10.15.1", arch: "x86_64", family: "mac"  
$
```

21. mvn -V package

This command prints the maven version and then continue with the build. It's equivalent to the commands `mvn -v;mvn package` .

22. mvn -DskipTests package

The `skipTests` system property is used to skip the unit test cases from the build cycle. We can also use `-Dmaven.test.skip=true` to skip the test cases execution.

23. mvn -T 4 package

This command tells maven to run parallel builds using the specified thread count. It's useful in multiple module projects where modules can be built in parallel. It can reduce the build time of the project.

Maven Commands Cheat Sheet



20+ MUST KNOW COMMANDS

1

mvn clean

Cleans the maven project by deleting the target directory.

3

mvn package

Build the maven project and create JAR, WAR files.

5

mvn deploy

Deploy the build artifact to the remote repository

7

mvn dependency:tree

Generates the dependency tree of the maven project.

9

mvn archetype:generate

Used to create a maven project from the archetype template project

11

mvn site:site

Generate a site for the maven project.

13

mvn compile

compile the source code of the project

15

mvn -f dir/pom.xml package

Force the use of an alternate POM file (or directory with pom.xml)

17

mvn -q package

runs the maven command in the quiet mode, only show errors and the test cases results.

2

mvn compiler:compile

Compiles the Java source classes. Use **"mvn compiler:testCompile"** to compile the test classes.

4

mvn install

Build the maven project and install the package files (JAR, WAR, pom.xml, etc) to the local repository.

6

mvn validate

validate the project is correct and all necessary information is available

8

mvn dependency:analyze

Analyze the maven project to identify the unused declared and used undeclared dependencies

10

mvn -help

Prints the usage and all the different options we can use with the mvn command

12

mvn test

test the compiled source code using a suitable unit testing framework.

14

mvn verify

run any checks on results of integration tests to ensure quality criteria are met

16

mvn -o package

runs the maven command in the offline mode.

17

mvn -X package

runs the build and produces output in the debug mode.

19

mvn -v

Display maven version information.

20

mvn -V package

Display maven version information and continue with the build

21

mvn -DskipTests package

skips running the test cases of the project, you can also use **-Dmaven.test.skip=true** option.

22

mvn -T 4 clean install

parallel build with 4 threads, useful to increase the build performance in the multiple module project.



© <https://www.journaldev.com>

Maven Commands Cheat Sheet