Also, as mentioned in the [first section](https://maven.apache.org/guides/introduction/introduction-to-the-pom.html#What_is_a_POM), if the configuration details are not specified, Maven will use their defaults. One of these default values is the packaging type. Every Maven project has a packaging type. If it is not specified in the POM, then the **default value "jar"** would be used.

<https://maven.apache.org/guides/introduction/introduction-to-the-pom.html>

The first execution with id "execution1" binds this configuration to the test phase. The second execution does not have a <phase> tag, how do you think will this execution behave? Well, goals can have a default phase binding as discussed further below. If the goal has a default phase binding then it will execute in that phase. But if the goal is not bound to any lifecycle phase then it simply won't be executed during the build lifecycle.

……

Now, let us have another mojo example which shows a default lifecycle phase binding.

1. /\*\*
2. \* @goal query
3. \* @phase package
4. \*/
5. public class MyBindedQueryMojo
6. extends AbstractMojo
7. {
8. /\*\*
9. \* @parameter expression="${query.url}"
10. \*/
11. private String url;
13. /\*\*
14. \* @parameter default-value="60"
15. \*/
16. private int timeout;
18. /\*\*
19. \* @parameter
20. \*/
21. private String[] options;
23. public void execute()
24. throws MojoExecutionException
25. {
26. ...
27. }
28. }

From the above mojo example, MyBindedQueryMojo is by default bound to the package phase (see the @phase notation). But if we want to execute this mojo during the install phase and not with package we can rebind this mojo into a new lifecycle phase using the <phase> tag under <execution>.

1. <project>
2. ...
3. <build>
4. <plugins>
5. <plugin>
6. <artifactId>maven-myquery-plugin</artifactId>
7. <version>1.0</version>
8. <executions>
9. <execution>
10. <id>execution1</id>
11. <phase>install</phase>
12. <configuration>
13. <url>http://www.bar.com/query</url>
14. <timeout>15</timeout>
15. <options>
16. <option>four</option>
17. <option>five</option>
18. <option>six</option>
19. </options>
20. </configuration>
21. <goals>
22. <goal>query</goal>
23. </goals>
24. </execution>
25. </executions>
26. </plugin>
27. </plugins>
28. </build>
29. ...
30. </project>

<https://maven.apache.org/guides/mini/guide-configuring-plugins.html>

**Supported Mojo Javadoc Tags**

1. import org.apache.maven.execution.MavenSession;
2. import org.apache.maven.plugin.AbstractMojo;
3. import org.apache.maven.plugin.MojoExecution;
4. import org.apache.maven.plugin.descriptor.PluginDescriptor;
5. import org.apache.maven.project.MavenProject;
6. import org.apache.maven.settings.Settings;
8. /\*\*
9. \* Mojo Description: goal is the only required annotation.
10. \*
11. \* @goal <goalName>
12. \* @aggregator
13. \* @configurator <roleHint>
14. \* @execute phase="<phaseName>" lifecycle="<lifecycleId>" goal="<goalName>"
15. \* @executionStrategy <once-per-session|always>
16. \* @inheritByDefault <true|false>
17. \* @instantiationStrategy <per-lookup|singleton|keep-alive|poolable>
18. \* @phase <phaseName>
19. \* @requiresDependencyResolution <compile|runtime|compile+runtime|test>
20. \* @requiresDependencyCollection <compile|runtime|compile+runtime|test> (since Maven 3.0)
21. \* @requiresDirectInvocation <false|true>
22. \* @requiresOnline <false|true>
23. \* @requiresProject <true|false>
24. \* @requiresReports <false|true> (unsupported since Maven 3.0)
25. \* @threadSafe (since Maven 3.0)
26. \* @since <since-text>
27. \* @deprecated <deprecated-text>
28. \*/
29. public class MyMojo
30. extends AbstractMojo
31. {
32. /\*\*
33. \* @parameter name="parameter" alias="myAlias" implementation="" property="aProperty" default-value="${anExpression}"
34. \* @readonly
35. \* @required
36. \* @since <since-text>
37. \* @deprecated <deprecated-text>
38. \*/
39. private String parameter;
41. /\*\*
42. \* @component role="..." roleHint="..."
43. \*/
44. private Component component;
46. // sample objects taken from Maven API through PluginParameterExpressionEvaluator
48. /\*\*
49. \* @parameter default-value="${session}"
50. \* @readonly
51. \*/
52. private MavenSession session;
54. /\*\*
55. \* @parameter default-value="${project}"
56. \* @readonly
57. \*/
58. private MavenProject project;
60. /\*\*
61. \* @parameter default-value="${mojoExecution}"
62. \* @readonly
63. \*/
64. private MojoExecution mojo;
66. /\*\*
67. \* @parameter default-value="${plugin}" // Maven 3 only
68. \* @readonly
69. \*/
70. private PluginDescriptor plugin;
72. /\*\*
73. \* @parameter default-value="${settings}"
74. \* @readonly
75. \*/
76. private Settings settings;
78. /\*\*
79. \* @parameter default-value="${project.basedir}"
80. \* @readonly
81. \*/
82. private File basedir;
84. /\*\*
85. \* @parameter default-value="${project.build.directory}"
86. \* @readonly
87. \*/
88. private File target;
90. public void execute()
91. {
92. ...
93. }
94. }

### Supported Mojo Javadoc Tags

1. import org.apache.maven.execution.MavenSession;
2. import org.apache.maven.plugin.AbstractMojo;
3. import org.apache.maven.plugin.MojoExecution;
4. import org.apache.maven.plugin.descriptor.PluginDescriptor;
5. import org.apache.maven.project.MavenProject;
6. import org.apache.maven.settings.Settings;
8. /\*\*
9. \* Mojo Description: goal is the only required annotation.
10. \*
11. \* @goal <goalName>
12. \* @aggregator
13. \* @configurator <roleHint>
14. \* @execute phase="<phaseName>" lifecycle="<lifecycleId>" goal="<goalName>"
15. \* @executionStrategy <once-per-session|always>
16. \* @inheritByDefault <true|false>
17. \* @instantiationStrategy <per-lookup|singleton|keep-alive|poolable>
18. \* @phase <phaseName>
19. \* @requiresDependencyResolution <compile|runtime|compile+runtime|test>
20. \* @requiresDependencyCollection <compile|runtime|compile+runtime|test> (since Maven 3.0)
21. \* @requiresDirectInvocation <false|true>
22. \* @requiresOnline <false|true>
23. \* @requiresProject <true|false>
24. \* @requiresReports <false|true> (unsupported since Maven 3.0)
25. \* @threadSafe (since Maven 3.0)
26. \* @since <since-text>
27. \* @deprecated <deprecated-text>
28. \*/
29. public class MyMojo
30. extends AbstractMojo
31. {
32. /\*\*
33. \* @parameter name="parameter" alias="myAlias" implementation="" property="aProperty" default-value="${anExpression}"
34. \* @readonly
35. \* @required
36. \* @since <since-text>
37. \* @deprecated <deprecated-text>
38. \*/
39. private String parameter;
41. /\*\*
42. \* @component role="..." roleHint="..."
43. \*/
44. private Component component;
46. // sample objects taken from Maven API through PluginParameterExpressionEvaluator
48. /\*\*
49. \* @parameter default-value="${session}"
50. \* @readonly
51. \*/
52. private MavenSession session;
54. /\*\*
55. \* @parameter default-value="${project}"
56. \* @readonly
57. \*/
58. private MavenProject project;
60. /\*\*
61. \* @parameter default-value="${mojoExecution}"
62. \* @readonly
63. \*/
64. private MojoExecution mojo;
66. /\*\*
67. \* @parameter default-value="${plugin}" // Maven 3 only
68. \* @readonly
69. \*/
70. private PluginDescriptor plugin;
72. /\*\*
73. \* @parameter default-value="${settings}"
74. \* @readonly
75. \*/
76. private Settings settings;
78. /\*\*
79. \* @parameter default-value="${project.basedir}"
80. \* @readonly
81. \*/
82. private File basedir;
84. /\*\*
85. \* @parameter default-value="${project.build.directory}"
86. \* @readonly
87. \*/
88. private File target;
90. public void execute()
91. {
92. ...
93. }
94. }

<https://maven.apache.org/plugin-tools/maven-plugin-tools-java/index.html>

### Lifecycle

clean lifecycle is defined directly with its plugin bindings.

<phases>

<phase>pre-clean</phase>

<phase>clean</phase>

<phase>post-clean</phase>

</phases>

<default-phases>

<clean>

org.apache.maven.plugins:maven-clean-plugin:2.5:clean

</clean>

</default-phases>

<https://maven.apache.org/ref/3.5.2/maven-core/lifecycles.html#clean_Lifecycle>