Summary

The Maven build follows the core build life cycle concept of the software project and defines the life cycle from build initialization to artifact distribution.

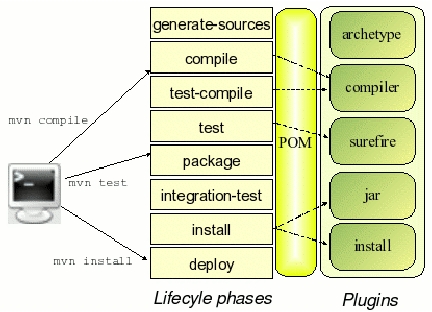
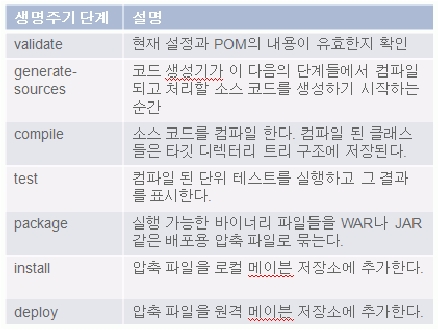
Description

The Maven life cycle phase is bound with plug-ins and the plug-ins execute commands. According to the life cycle, builds are carried out sequentially to carry out the command executed by the developer.

Example)When the mvn install command is executed, the install command is executed by using compile and test commands from the generate-sources phase.

Example) Execute the Java comple: $mvn complile command.

Maven 2 Basic Life Cycle Phase



life cycle phase

validate

initialize

generate-

sources

Description

Check if the current settings and POM are valid. The process verifies POM.XML file tree structure.

The moment when initialization is allowed before main tasks to be carried out in the build cycle

The moment when the code generator is complied in the next phases to generate source codes

process-sources Provide the parsing, modification and change of sources. Both normal and generated codes are handled here.

generate-

resources

process-

resources

compile

process-classes

generate-test-

sources

process-test-

sources

generate-test-

resources

process-test-

resources

test-compile

test

package

pre-integration-

test

integration-test

post-integration-

test

verify

install

deploy

The phase where resources are generated. Usually meta data files and config files are included.

The resource files are handled, instead of the source codes in the previous phase. Resource files are modified, changed and rearranged.

The source files are compiled. The compiled classes are stored in the target directory tree structure.

Class file conversion and revision phase are handled. Byte code weaver and instrument tool are operated

The moment Mojo operates, which creates the unit test codes.

Carry out the tasks required for test source code before compiling. In this phase, the source code can be modified, converted and copied.

The creation of rest related resource is allowed.

The handling, conversion and rearrangement related with test are allowed.

The unit test source code is compiled.

Execute the compiled unit test and display the results.

Use a compression file such as JAR and WAR to bundle executable binary files.

Prepare the integrated test which is to test the code in the actual deployment environment. In this phase, the above bundled compressed files can be deployed in the server

Carry out the actual integrated test.

Disable the preparation status of integrated test. This can include the reset or re-initialization of the test environment.

Verify the integrity and validity of deployable compressed files. After this process, compressed files are installed

Add compressed files to the local maven directory ,which makes the user to be able to use other modules that depends on it.

Add compressed files to the remote maven directory, which allow more users to be able to use this artifact.

Environment configuration

Basically, the Maven plug-in is included in the Maven distribution file, requiring no additional Maven plug-ins .

If you have to use a plug-in which is not provided, check the Maven plug-in list [http://maven.apache.org/plugins/] and add it.

Manual

1. To carry out all build operations defined in the life cycle phase, execute the deploy command, the last phase.

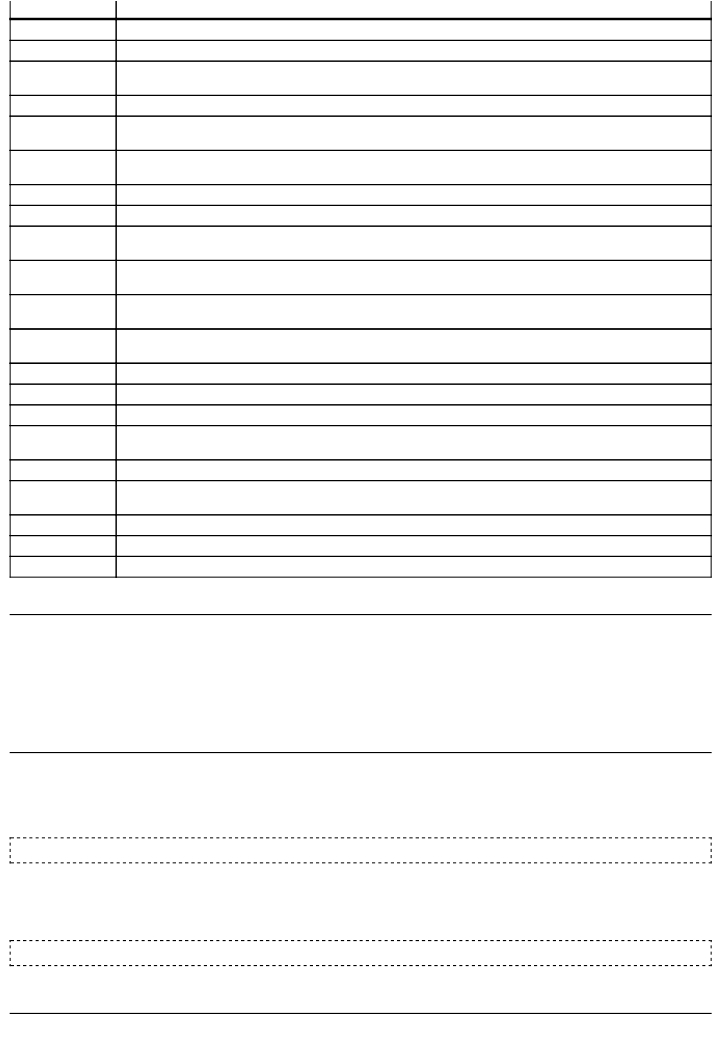
The deploy command is executed after validate phase and then install phase are carried out.

mvn deploy

2. Two or more maven commands can be executed sequentially. The following command is usually used to delete the content below the Target directory and carry out a new build in order to distribute the package into the local repository.

mvn clean install

References



Maven Build Lifecycle [http://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html]