
DevOps Shack

100 Maven Errors, Solution and RCA

1. Error: Unable to find valid certification path to requested target

Root Cause:

- Maven is trying to connect to a remote repository over HTTPS, but the SSL certificate is not trusted.

Solution:

Add the missing certificate to the Java keystore:

```
keytool -import -trustcacerts -keystore cacerts  
-storepass changeit -alias mycert -file mycert.crt
```

Alternatively, bypass SSL verification:

```
mvn -Dmaven.wagon.http.ssl.insecure=true clean install
```

2. Error: Could not resolve dependencies for project

Root Cause:

- The specified dependencies are not available in the configured Maven

repository.

Solution:

Run:

mvn dependency:tree

- to identify the missing dependencies.
- If using an internal repository, check the **.m2/settings.** file for correct repository configuration.

3. Error: Plugin execution not covered by lifecycle configuration

Root Cause:

- The plugin is missing execution configuration in **pom..**

Solution:

Add the required **<executions>** block:

```
<executions>
  <execution>
    <goals>
      <goal>compile</goal>
    </goals>
  </execution>
</executions>
```

4. Error: Out of Memory



Root Cause:

- Maven is running out of allocated memory due to a large project or dependencies.

Solution:

Increase JVM memory:

```
export MAVEN_OPTS="-Xmx1024m -XX:MaxPermSize=512m"
```

Use parallel builds:

```
mvn clean install -T 1C
```

5. Error: Build failure due to missing SNAPOT dependency

Root Cause:

- The dependency is a **SNAPOT** version that is either unavailable or not updated.

Solution:

Force update dependencies:

```
mvn -U clean install
```

- Verify the repository URL in **settings**.

6. Error: No compiler is provided in this environment

Root Cause:



- JDK is not installed or **JAVA_HOME** is misconfigured.

Solution:

Install JDK and set **JAVA_HOME**:

```
export JAVA_HOME=/path/to/jdk
```

7. Error: Cannot find symbol in Java files

Root Cause:

- Missing or incorrect dependencies.

Solution:

- Check and fix dependencies in **pom..**

Ensure correct Java version:

```
<properties>
  <maven.compiler.source>1.8</maven.compiler.source>
  <maven.compiler.target>1.8</maven.compiler.target>
</properties>
```

8. Error: Failure to find artifact

Root Cause:

- The artifact is missing in the configured repository.

Solution:



Try:

```
mvn dependency:purge-local-repository
```

9. Error: Maven could not transfer artifact

Root Cause:

- Firewall or proxy blocking repository access.

Solution:

Configure proxy in `~/.m2/settings.:`

```
<proxies>
  <proxy>
    <id>proxy</id>
    <active>true</active>
    <protocol>http</protocol>
    <host>proxy.example.com</host>
    <port>8080</port>
  </proxy>
</proxies>
```

10. Error: Maven Central Repository not reachable

Root Cause:

- Network issue or incorrect repository URL.

Solution:



Check if Maven Central is accessible:

`curl https://repo.maven.apache.org/maven2/`

If blocked, use a mirror:

```
<mirrors>
  <mirror>
    <id>maven-central</id>
    <url>https://repo.maven.apache.org/maven2</url>
    <mirrorOf>central</mirrorOf>
  </mirror>
</mirrors>
```

11. Error: Dependency version conflict

Root Cause:

- Two or more dependencies use different versions of the same artifact.

Solution:

- Use `mvn dependency:tree` to identify conflicts.
- Explicitly declare the required version in `pom..`

12. Error: Unable to load class

Root Cause:

- A required class is missing in the dependencies.



Solution:

Run:

mvn dependency:analyze

- to check for missing dependencies.

13. Error: Compilation failure in tests

Root Cause:

- Missing test dependencies.

Solution:

Ensure **pom.** includes:

```
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <scope>test</scope>
</dependency>
```

14. Error: Maven Outdated Plugin Version

Root Cause:

- The plugin version is old or deprecated.

Solution:



- Update the plugin to the latest version.

15. Error: The specified POM does not exist

Root Cause:

- The **pom.** file is missing or incorrectly named.

Solution:

Verify the file path:

```
ls pom.
```

16. Error: Maven build stuck at downloading dependencies

Root Cause:

- Slow or unresponsive repository.

Solution:

Use a mirror repository:

```
<mirror>
  <id>fast-mirror</id>
  <url>https://myfastrepo.com/maven2</url>
  <mirrorOf>central</mirrorOf>
</mirror>
```

17. Error: Checksum validation failed

Root Cause:



- Corrupt dependency file.

Solution:

Delete corrupted files:

```
rm -rf ~/.m2/repository
```

- Run `mvn clean install` again.

18. Error: Maven JAR Signing Failure

Root Cause:

- Incorrect GPG key configuration.

Solution:

Configure `settings.`:

```
<servers>
  <server>
    <id>gpg-passphrase</id>
    <passphrase>your-passphrase</passphrase>
  </server>
</servers>
```

19. Error: Tests fail due to missing database

Root Cause:



- Test cases require a running database.

Solution:

- Use an embedded database like H2 or configure a test database.

20. Error: Maven goals not recognized

Root Cause:

- Incorrect Maven installation.

Solution:

Check installation:

```
mvn -version
```

21. Error: Failed to read artifact descriptor for xyz:jar:version

Root Cause:

- A transitive dependency is missing or corrupt in the repository.

Solution:

Run:

```
mvn dependency:purge-local-repository
```

Delete the dependency from **.m2** and redownload:

```
rm -rf ~/.m2/repository/xyz
mvn clean install
```

22. Error: No plugin found for prefix 'xyz'

Root Cause:

- The specified Maven plugin is missing or not installed.

Solution:

Ensure the plugin is declared in **pom.**:

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-xyz-plugin</artifactId>
  <version>latest-version</version>
</plugin>
```

Run:

```
mvn help:describe -Dplugin=xyz
```

23. Error: Could not find or load main class

Root Cause:

- The main class is not defined or misconfigured.



Solution:

Ensure **pom.** has:

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <configuration>
        <archive>
          <manifest>

<mainClass>com.example.Main</mainClass>
          </manifest>
        </archive>
      </configuration>
    </plugin>
  </plugins>
</build>
```

24. Error: Invalid target release

Root Cause:

- Java version mismatch.

Solution:



Specify Java version in **pom.**:

```
<properties>
  <maven.compiler.source>11</maven.compiler.source>
  <maven.compiler.target>11</maven.compiler.target>
</properties>
```

25. Error: **Unsupported major.minor version**

Root Cause:

- The project is compiled with a different Java version than the runtime.

Solution:

Run:

```
java -version
javac -version
```

26. Error: **Could not create parent directory for .m2/repository**

Root Cause:

- The user running Maven does not have permission to create folders.

Solution:

Change ownerip:

```
sudo chown -R $(whoami) ~/.m2
```



27. Error: **Unknown lifecycle phase**

Root Cause:

- Invalid Maven command.

Solution:

Run:

```
mvn help:describe -Dcmd=yourCommand
```

28. Error: **Artifact has not been packaged yet**

Root Cause:

- Maven requires **package** before **install**.

Solution:

Run:

```
mvn clean package install
```

29. Error: **Maven fails with Exit Code 1**

Root Cause:

- General failure in the build process.

Solution:

Run:



```
mvn -X clean install
```

- to enable debug mode

30. Error: **Could not calculate build plan**

Root Cause:

- An unresolved plugin version.

Solution:

- Add explicit plugin versions in **pom**.

31. Error: **Checksum validation failed**

Root Cause:

- A corrupt artifact was downloaded.

Solution:

Delete the artifact and re-download:

```
rm -rf ~/.m2/repository/com/example  
mvn clean install
```

32. Error: **Invalid repository URL**

Root Cause:

- The repository URL is incorrect in **settings**..

Solution:

- Verify the repository URL by testing it in a browser.

33. Error: **Could not find goal 'xyz'**

Root Cause:

- A required plugin is missing.

Solution:

Run:

```
mvn help:describe -Dplugin=xyz
```

34. Error: **Maven build timeout**

Root Cause:

- Slow network or unresponsive repository.

Solution:

Increase timeout:

```
mvn -Dmaven.wagon.httpconnectionTimeout=50000 clean  
install
```

35. Error: **File Not Found Exception in target directory**

Root Cause:



- The build process deleted the required file.

Solution:

Run:

```
mvn clean compile
```

36. Error: **Could not find metadata**

Root Cause:

- Maven metadata is missing.

Solution:

Refre metadata:

```
mvn -U clean install
```

37. Error: **Failure to transfer xyz from repository**

Root Cause:

- The repository is down or blocked.

Solution:

- Try a different mirror.

38. Error: **Parent POM is missing**

Root Cause:



- The parent project is not installed locally.

Solution:

Install the parent project first:

```
mvn install -N
```

39. Error: **Invalid in POM**

Root Cause:

- A syntax error in **pom..**

Solution:

Validate with:

```
mvn validate
```

40. Error: **Unable to find Maven executable**

Root Cause:

- Maven is not installed or not in **PATH**.

Solution:

Run:

```
which mvn
```

41. Error: **Maven running as root warning**

Root Cause:

- Running Maven as **root** is unsafe.

Solution:

- Run as a normal user.

42. Error: **org.apache.maven.lifecycle.LifecycleExecutionException**

Root Cause:

- Build failure due to an execution issue.

Solution:

Use:

```
mvn -e clean install
```

43. Error: **Project packaging type is missing**

Root Cause:

- **pom.** is missing **<packaging>**.

Solution:

Add:

```
<packaging>jar</packaging>
```



44. Error: **Invalid proxy configuration**

Root Cause:

- Incorrect proxy setup.

Solution:

- Update **settings..**

45. Error: **Build failure in multi-module project**

Root Cause:

- A submodule is failing.

Solution:

Build individual modules:

```
mvn clean install -pl module-name
```

46. Error: **Access denied to repository**

Root Cause:

- Incorrect credentials.

Solution:

- Update **settings.** with correct **<server>** credentials.

47. Error: **Java heap space**



Root Cause:

- Insufficient memory.

Solution:

Increase:

```
export MAVEN_OPTS="-Xmx2048m"
```

48. Error: **Could not resolve classpath dependency**

Root Cause:

- A missing JAR file.

Solution:

Run:

```
mvn dependency:resolve
```

49. Error: **No connector available**

Root Cause:

- Maven needs **wagon-http**.

Solution:

Add:

```
<dependency>  
  <groupId>org.apache.maven.wagon</groupId>
```

```
<artifactId>wagon-http</artifactId>
</dependency>
```

50. Error: Javadoc errors breaking build

Root Cause:

- JavaDoc linting failed.

Solution:

Use:

```
mvn clean install -DskipTests -Dskip.javadoc=true
```

51. Error: Could not resolve archetype

Root Cause:

- The specified Maven archetype is not available in the repository.

Solution:

Update the local archetype catalog:

```
mvn archetype:generate -U
```

- Manually specify a repository in `pom..`

52. Error: Plugin execution not allowed

Root Cause:



- The plugin is blocked by a security policy.

Solution:

Allow execution by adding:

```
<pluginManagement>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>

      <artifactId>maven-compiler-plugin</artifactId>
      <version>latest-version</version>
    </plugin>
  </plugins>
</pluginManagement>
```

53. Error: Profile cannot be activated because it does not exist

Root Cause:

- A profile is referenced but not defined.

Solution:

Ensure the profile exists in **pom.**:

```
<profiles>
```



```
<profile>
  <id>my-profile</id>
  <activation>
    <activeByDefault>true</activeByDefault>
  </activation>
</profile>
</profiles>
```

54. Error: The specified command is invalid

Root Cause:

- Typo in the Maven command.

Solution:

Check available commands:

```
mvn help:describe -Dcmd=<command>
```

55. Error: Unable to delete target directory

Root Cause:

- Files in the **target** directory are locked.

Solution:

Force delete:

```
rm -rf target/
```



- Restart the system if necessary.

56. Error: Project dependencies include references to the local repository

Root Cause:

- A dependency was installed locally but is missing from a ared repository.

Solution:

Deploy it to a ared repository:

`mvn deploy`

57. Error: Nexus repository unreachable

Root Cause:

- The Nexus server is down or misconfigured.

Solution:

- Verify URL and credentials in `settings..`

58. Error: Failure to find parent POM

Root Cause:

- The parent POM is missing in the local repository.

Solution:

Install the parent POM manually:



```
mvn install -N
```

59. Error: Unrecognized option for the Maven compiler plugin

Root Cause:

- The wrong compiler plugin version is used.

Solution:

Upgrade:

```
<plugin>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>latest-version</version>
</plugin>
```

60. Error: Unsupported class version

Root Cause:

- The compiled class is not compatible with the Java runtime.

Solution:

Ensure correct Java version:

```
<maven.compiler.source>11</maven.compiler.source>
<maven.compiler.target>11</maven.compiler.target>
```

61. Error: Could not resolve plugin version



Root Cause:

- The plugin version is missing.

Solution:

- Specify the version in **pom..**

62. Error: **Compilation failure due to missing JAR**

Root Cause:

- A required JAR file is missing.

Solution:

Run:

```
mvn dependency:tree  
mvn dependency:resolve
```

63. Error: **Mojo execution failed**

Root Cause:

- The plugin execution encountered an issue.

Solution:

Debug with:

```
mvn -X
```



64. Error: **Missing artifact descriptor**

Root Cause:

- The artifact descriptor is missing.

Solution:

Update repositories:

```
mvn clean install -U
```

65. Error: **Parent project has wrong packaging type**

Root Cause:

- The parent POM is declared incorrectly.

Solution:

- Use `<packaging>pom</packaging>` for parent POMs.

66. Error: **Could not find Maven executable**

Root Cause:

- Maven is not installed or misconfigured.

Solution:

- Reinstall Maven.

67. Error: **Dependency resolution failed due to incorrect scope**

Root Cause:

- A dependency is used incorrectly.

Solution:

Adjust scope:

`<scope>provided</scope>`

68. Error: **Failure to transfer checksum**

Root Cause:

- A checksum validation error occurred.

Solution:

Retry with:

`mvn clean install -U`

69. Error: **Could not resolve build extensions**

Root Cause:

- A required build extension is missing.

Solution:

- Check `<build>` section in `pom..`

70. Error: **Plugin requires additional configuration**



Root Cause:

- A plugin is missing required parameters.

Solution:

- Check plugin documentation.

71. Error: **Failure to determine project location**

Root Cause:

- The **pom.** file is missing.

Solution:

- Ensure you are inside the project directory.

72. Error: **Compilation fails on specific OS**

Root Cause:

- Platform-dependent issue.

Solution:

Set encoding:

```
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
```

73. Error: **Failure to find Maven home**



Root Cause:

- The **MAVEN_HOME** variable is incorrect.

Solution:

Set:

```
export MAVEN_HOME=/path/to/maven
```

74. Error: **Multiple versions of dependency found**

Root Cause:

- Version conflict.

Solution:

- Use **<dependencyManagement>**.

75. Error: **Module not found in multimodule build**

Root Cause:

- The module is missing.

Solution:

- Ensure it is declared in **<modules>**.

76. Error: **Circular dependency detected**

Root Cause:



- Two modules depend on each other.

Solution:

- Break the circular dependency.

77. Error: **Unable to resolve JAR in a different repository**

Root Cause:

- Repository misconfiguration.

Solution:

- Check repository URLs.

78. Error: **Skipping deployment due to missing credentials**

Root Cause:

- The credentials for deployment are missing.

Solution:

- Add credentials in **settings..**

79. Error: **Failure to determine project packaging**

Root Cause:

- The **<packaging>** tag is missing.

Solution:



- Add `<packaging>jar</packaging>`.

80. Error: **Failure due to corrupted repository**

Root Cause:

- A corrupt `.m2` repository.

Solution:

Delete:

```
rm -rf ~/.m2/repository
```

81. Error: **Cannot access repository**

Root Cause:

- The configured repository is either down, blocked by a firewall, or requires authentication.

Solution:

Check if the repository URL is correct:

```
curl -I https://repo.maven.apache.org/maven2/
```

If authentication is needed, add credentials to `~/.m2/settings.`:

```
<servers>
  <server>
```



```
<id>your-repo-id</id>
<username>your-username</username>
<password>your-password</password>
</server>
</servers>
```

82. Error: **Maven execution hangs indefinitely**

Root Cause:

- A slow or unresponsive repository, infinite loops in a plugin, or network issues.

Solution:

Run with debug logs:

```
mvn -X clean install
```

Try skipping slow operations:

```
mvn clean install -Dmaven.javadoc.skip=true
```

Check active processes:

```
ps aux | grep maven
```

83. Error: **Skipping deployment due to missing credentials**

Root Cause:



- Deployment to a private repository failed due to missing credentials.

Solution:

Ensure the credentials are correctly added to **settings.**:

```
<server>
  <id>my-repo</id>
  <username>admin</username>
  <password>mypassword</password>
</server>
```

84. Error: **Missing resource file**

Root Cause:

- A required file in **src/main/resources** or **src/test/resources** is missing.

Solution:

- Verify that the file exists and is correctly referenced.

85. Error: **Maven build fails due to missing class in JAR**

Root Cause:

- The required class is missing from the packaged JAR.

Solution:

Run:



```
mvn dependency:tree
```

Rebuild the JAR:

```
mvn clean package
```

86. Error: Maven wrapper (mvnw) issues

Root Cause:

- The Maven wrapper script (**mvnw**) is corrupted or missing.

Solution:

Recreate the wrapper:

```
mvn -N io.takari:maven:wrapper
```

If using Windows, ensure **mvnw.cmd** has execution permissions:

```
chmod +x mvnw
```

87. Error: Multiple main classes found

Root Cause:

- There are multiple classes with a **main** method.

Solution:

Specify the main class in **pom.**:



```
<manifest>
  <mainClass>com.example.MainClass</mainClass>
</manifest>
```

88. Error: Build fails in multi-module project

Root Cause:

- One of the submodules fails, stopping the entire build.

Solution:

Skip failing modules:

```
mvn clean install -pl module-name -am
```

Use **-T 1C** for parallel builds:

```
mvn clean install -T 1C
```

89. Error: Maven build fails due to missing transitive dependency

Root Cause:

- A required transitive dependency is not found.

Solution:

Run:

```
mvn dependency:tree
```



`mvn dependency:resolve`

- Explicitly include the missing dependency in `pom..`

90. Error: `Maven release plugin fails`

Root Cause:

- The release process is interrupted or misconfigured.

Solution:

Run:

`mvn release:clean release:prepare release:perform`

- Ensure versioning is correct in `pom..`

91. Error: `Javadoc generation fails`

Root Cause:

- Javadoc warnings are treated as errors.

Solution:

Disable strict Javadoc checking:

`mvn clean install -Dskip.javadoc=true`

92. Error: `Tests fail due to incorrect test runner`

Root Cause:



- JUnit version mismatch.

Solution:

Ensure correct JUnit version:

```
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <scope>test</scope>
</dependency>
```

93. Error: Jar file contains duplicate entries

Root Cause:

- Multiple dependencies bring the same classes.

Solution:

Use the **maven-ade-plugin** to handle duplicates:

```
<plugin>
  <artifactId>maven-ade-plugin</artifactId>
  <executions>
    <execution>
      <phase>package</phase>
      <goals>
        <goal>ade</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```



```
        </goals>
      </execution>
    </executions>
  </plugin>
```

94. Error: **Incorrect POM file format**

Root Cause:

- Syntax errors in **pom..**

Solution:

Validate the format:

```
mvn validate
```

95. Error: **Cannot find project dependencies in the local repository**

Root Cause:

- Dependencies are not installed locally.

Solution:

Run:

```
mvn install
```

96. Error: **Dependency resolution fails due to multiple**



repositories

Root Cause:

- Conflicts between repositories.

Solution:

Define a **<mirror>** to ensure a single source:

```
<mirrors>
  <mirror>
    <id>central-mirror</id>
    <mirrorOf>central</mirrorOf>
    <url>https://repo.maven.apache.org/maven2</url>
  </mirror>
</mirrors>
```

97. Error: Plugin execution hangs due to infinite loop

Root Cause:

- Recursive execution of a plugin.

Solution:

- Avoid circular dependencies in **pom..**

98. Error: **Failed to execute goal**
org.codehaus.mojo:exec-maven-plugin



Root Cause:

- The **exec-maven-plugin** cannot find the required command.

Solution:

Ensure the correct goal is specified:

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>exec-maven-plugin</artifactId>
  <executions>
    <execution>
      <goals>
        <goal>java</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

99. Error: Maven build fails with a NullPointerException

Root Cause:

- A plugin is missing or has a bug.

Solution:

Run in debug mode:



```
mvn -X clean install
```

100. Error: **Unsupported POM version**

Root Cause:

- An outdated Maven version is being used.

Solution:

Upgrade Maven:

```
mvn -version
```

- Update the POM to use a compatible version.