

CodeArts Artifact

FAQs

Issue 01
Date 2024-10-18



Copyright © Huawei Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:

<https://www.huawei.com/en/psirt/vul-response-process>

For vulnerability information, enterprise customers can visit the following web page:

<https://securitybulletin.huawei.com/enterprise/en/security-advisory>

Contents

1 Release Repo.....	1
1.1 Why Can't I Upload Files or Create Directories on the Release Repos Homepage?.....	1
1.2 Can I Change the Dependency ID in pom.xml to Invoke a JAR File in My Release Repo?.....	1
1.3 Can I Restore Files in the Recycle Bin of My Release Repos?.....	1
2 Self-Hosted Repo.....	3
2.1 How Do I Upload Snapshots to a Maven Repository?.....	3
2.2 How Do I Pull Components from a Maven Repository?.....	6
2.3 Can I Call Software Packages in Self-Hosted Repos During Local Builds?.....	6
2.4 What Should I Do With Error Code 500 When Uploading Maven Package for a Gradle Build?.....	6
2.5 Why Can't the Repository Receive Requests?.....	7
2.6 Why Did the Dependency WAR or JAR Files Fail to Be Downloaded?.....	8
2.7 Why Is Error 401 Returned When Uploading Maven Components to Self-Hosted Repos?.....	8

1 Release Repo

1.1 Why Can't I Upload Files or Create Directories on the Release Repos Homepage?

The top-level directory names on this page map to the names of your projects that hold each software package.

You can only browse files and directories in this directory.

Click a project name, and upload files and create directories there.

1.2 Can I Change the Dependency ID in pom.xml to Invoke a JAR File in My Release Repo?

No.

Packages in Release Repos are used for deployment, not as dependencies during build.

You need to first upload your dependency to a self-hosted repo.

1.3 Can I Restore Files in the Recycle Bin of My Release Repos?

Symptom

A file cannot be restored from the recycle bin page. A message indicating that **duplicate file exists** is displayed.

Cause Analysis

A file with the same name exists at the restored location in the repository.

Solution

You can choose **Move and replace**, **Do not move**, or **Move and rename**.

- **Move and replace:** The file restored from the recycle bin will replace the file with the same name in the restored location.
- **Do not move:** Ignore the file restoration in the recycle bin.
- **Move and rename:** Both the original file and the file in the recycle bin are retained in the restored location. The file restored from the recycle bin will be renamed.

2 Self-Hosted Repo

2.1 How Do I Upload Snapshots to a Maven Repository?

Background

Snapshots can be uploaded in any of the following ways:

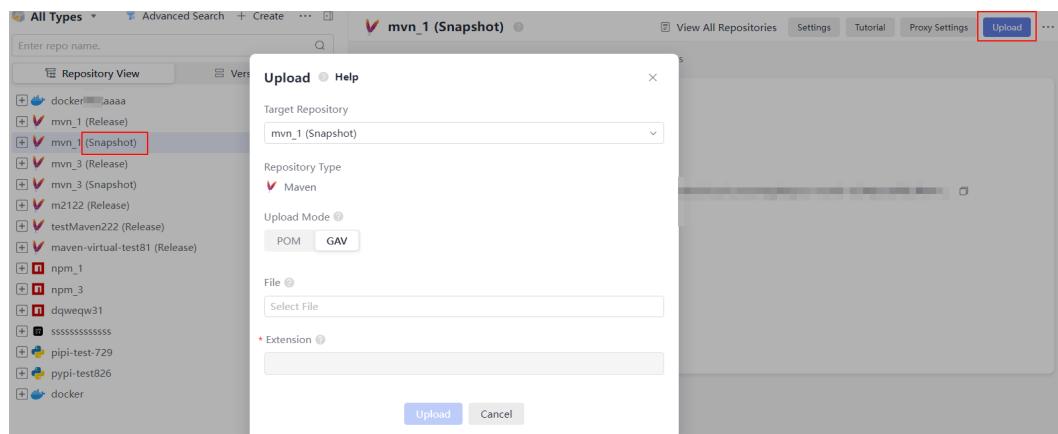
- [Uploading Snapshots on the Release Repos Page](#)
- [Uploading Snapshots Using the Maven CLI](#)
- [Releasing Snapshots to a Maven Repository Through CodeArts Build](#)

Uploading Snapshots on the Release Repos Page

Step 1 Log in to CodeArts.

Step 2 Choose **Services > Artifact**, click the **Self-hosted Repos** tab, and find the target repository.

Step 3 Click the Snapshot repository in the repository list. Click **Upload**. In the displayed dialog box, select **GAV**.



There are two GAV definition modes.

GAV Definition Mode	Description
POM	GAV information is extracted from POM files.
GAV	GAV information is manually specified.

Step 4 Set related parameters as prompted and upload the package.

----End

Uploading Snapshots Using the Maven CLI

Step 1 Select Maven as the package type, and choose the **Snapshot** repository in the repository list.

Step 2 Click **Tutorial** in the upper right corner.


Step 3 Configure the local Maven tool by following the configuration guide.

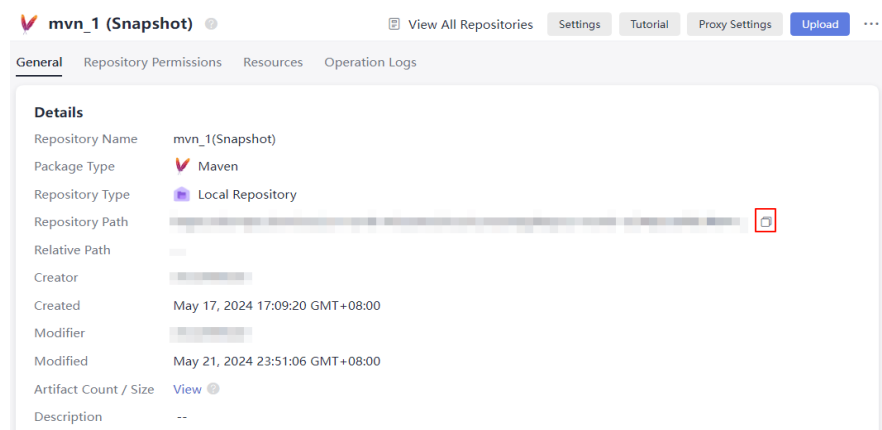
Step 4 Run **mvn deploy** to upload the Maven project.

In the Maven CLI, access the directory where the **pom.xml** file of the Maven project is stored, then run the following command to upload a local JAR package:

```
mvn deploy:deploy-file -DgroupId=com.huawei -DartifactId=aopalliance -Dversion=1.0-SNAPSHOT -Dpackaging=jar -Dfile=D:\aopalliance-1.0-SNAPSHOT.jar -Durl={Maven Snapshot address} -DrepositoryId=snapshots
```

NOTE

- Set **DgroupId**, **DartifactId**, **Dversion**, and **Dpackaging** as required.
- Set **Dfile** to the absolute path of the local JAR package.
- Set **Durl** to the Maven snapshot path, which can be obtained by clicking  in the following figure.



----End

Releasing Snapshots to a Maven Repository Through CodeArts Build

Step 1 Go to Repo, open the **pom.xml** file, and define the GAV information of the component to be uploaded.

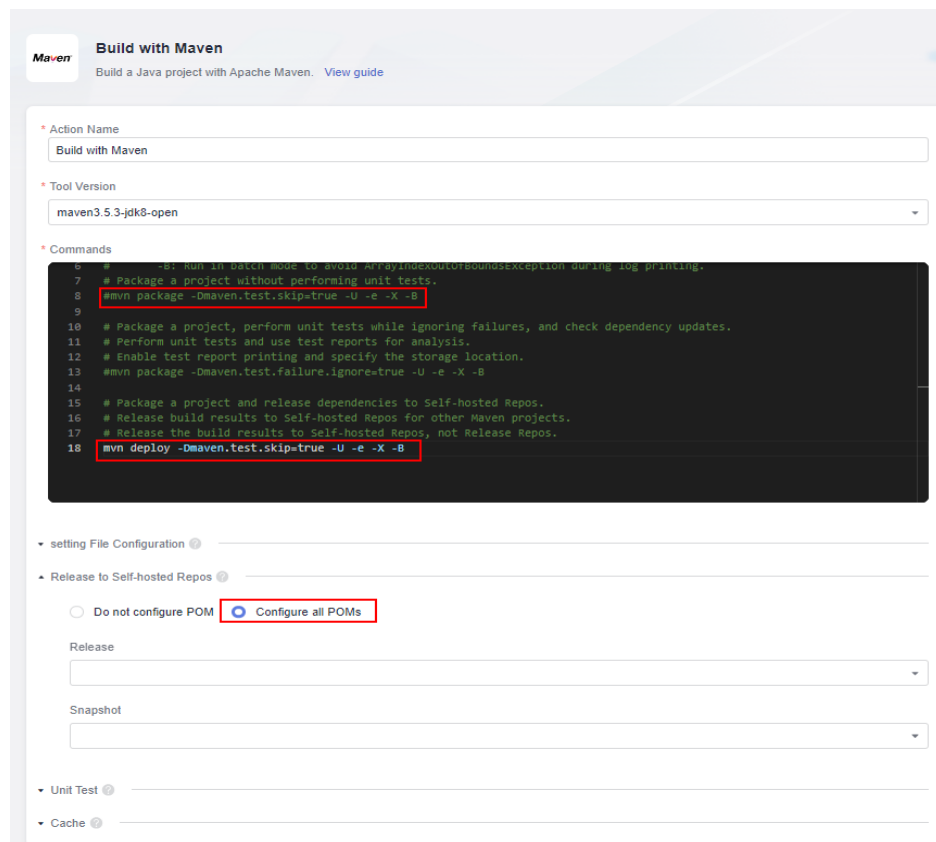


NOTE

- When a build task is run, CodeArts Build identifies the component properties uploaded to the Maven repository based on the definition.
- **version:** Releases are uploaded by default. To upload a Snapshot, add the suffix **-SNAPSHOT** to the value of **version**, for example, **1.0-SNAPSHOT**.

Step 2 Edit a build task in the build action **Build with Maven:**

- In the command box, comment out the **mvn package** command (add # before the command) and uncomment the **mvn deploy** command (delete # before the command).
- Click **Release to Self-hosted Repos**, and select **Configure all POMs**.



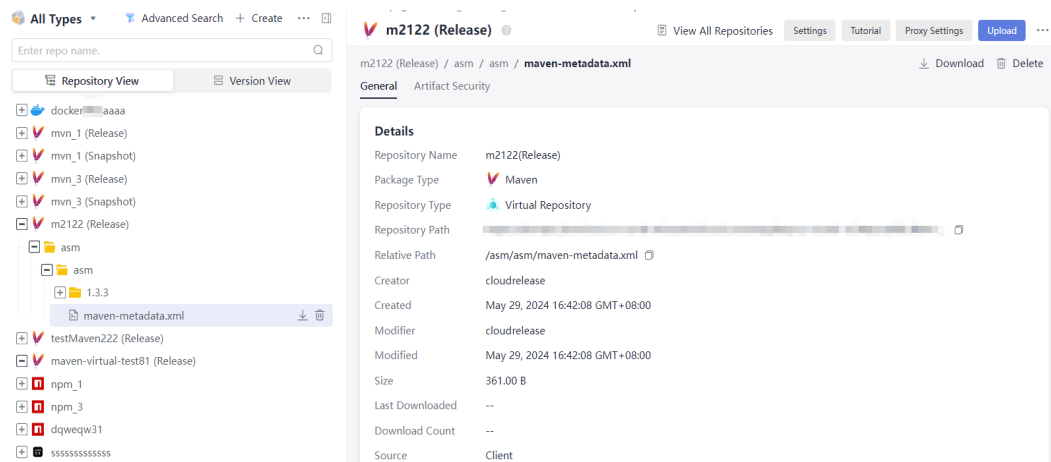
Step 3 Run a build task.

After the build task is executed, you can find the generated Maven component in the Maven repository.

----End

2.2 How Do I Pull Components from a Maven Repository?

Step 1 Go to the self-hosted repo page, and click the name of the component to be pulled. The **General** page is displayed.



Step 2 Obtain the dependency download address, copy it, and paste it to the **pom.xml** file.

----End

2.3 Can I Call Software Packages in Self-Hosted Repos During Local Builds?

Yes.

Go to the repository where the packages are stored and click **Tutorial** in the upper right corner. Download the configuration file and modify it by following the guide.

2.4 What Should I Do With Error Code 500 When Uploading Maven Package for a Gradle Build?

Symptom

A build task fails, and the log information similar to the following is displayed.

```
> Task :publishMavenPublicationToMavenRepository FAILED
FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':publishMavenPublicationToMavenRepository'.
> Failed to publish publication 'maven' to repository 'maven'
  > Could not write to resource 'https://devrepo.devcloud.huaweicloud.com/03/nexus/content/repositories/323aaa81851e42978ca4d3e7017ea94c_1_0/com/zzb/zzbFrame/0.0.1-SNAPSHOT/zzbFrame-0.0.1-20181025.085851-1.jar'.
  > Could not PUT 'https://devrepo.devcloud.huaweicloud.com/03/nexus/content/repositories/323aaa81851e42978ca4d3e7017ea94c_1_0/com/zzb/zzbFrame/0.0.1-SNAPSHOT/zzbFrame-0.0.1-20181025.085851-1.jar'. Received status code 500 from server:
```

Cause Analysis

The release address, rather than the snapshot address, is set.

Solution

Change the address to the snapshot address and upload the package again.

2.5 Why Can't the Repository Receive Requests?

Symptom

Local build task fails, **Connection reset** is displayed, and the log information similar to the following is displayed.

```
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-deploy-plugin:2.7:deploy (default-deploy) on project
base-parent: Failed to retrieve remote metadata [REDACTED] framework:base-parent:4.1.200-SNAPSHOT/maven-
metadata.xml: Could not transfer metadata [REDACTED] framework:base-parent:4.1.200-SNAPSHOT/maven-
metadata.xml from/to snapshots [REDACTED]
[REDACTED]
[REDACTED] framework/bas
e-parent/4.1.200-SNAPSHOT/maven-metadata.xml. Connection reset -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
```

Cause Analysis

The Java version is too early and does not support TLS 1.2.

Solution

- If Java 6 is used, upgrade it to Java 8 or later.
- If Java 7 is used, TLS 1.2 is supported. However, TLS 1.2 is not supported in versions earlier than 1.7.0_131-b31. You can run the following command to enable TLS 1.2:

```
mvn -Dhttps.protocols= TLSv1.2 <goals>
```

You can also add the following command to your environment or build script.

```
export MAVEN_OPTS=-Dhttps.protocols= TLSv1.2
```

2.6 Why Did the Dependency WAR or JAR Files Fail to Be Downloaded?

Symptom

The local tools cannot download components in the self-hosted repo. A message is displayed indicating that the POM file cannot be found. The log information similar to the following is recorded.

```
3: Task failed with an exception.
-----
* What went wrong:
Execution failed for task ':app:mergeXichangDebugResources'.
> Could not resolve all files for configuration ':app:xichangDebugRuntimeClasspath'.
   > Could not find com.pandabus.android:PBPullToRefresh-Library:1.8.3.
      Searched in the following locations:
         - https://[redacted]devcloud.huaweicloud.com/
           PBPullToRefresh-Library-1.8.3.pom
```

Cause Analysis

The POM file is missing in the dependency.

When downloading dependencies using Gradle or Maven, you need to download a POM file first, and then a JAR or WAR file. Otherwise, the download will fail.

Solution

Re-upload the components that cannot be downloaded according to the components uploading standard.

2.7 Why Is Error 401 Returned When Uploading Maven Components to Self-Hosted Repos?

Symptom

Failed to upload Maven components to self-hosted repos from the local IDE, and **401-Insufficient Permission** is displayed.

Cause Analysis

The self-hosted repo information configured in the **pom.xml** file does not match the **settings.xml** file.

Solution

When uploading components, replace the **repository_id** value in the **distributionManagement** element of the **pom.xml** file with the **repository_id** value in the **server** element of the **settings.xml** file.

The uploading process is as follows:

Step 1 Go to the self-hosted repo page, and choose Maven from the repository list.

Step 2 Click **Tutorial** in the upper right corner.

Step 3 Configure the local Maven tool by following the configuration guide.

Step 4 Run **mvn deploy** to upload the Maven project.

1. In the Maven CLI, access the directory where the **pom.xml** file of the Maven project is stored, check whether the **repository_id** value in the **distributionManagement** element of the **pom.xml** file matches the **repository_id** value in the **server** element of the **settings.xml** file.

setting.xml

```
<server>
  <id>release_cn-north-1</id>
  <username>cn-north-1</username>
  <password></password>
</server>
<server>
  <id>snapshot_cn-north-1</id>
  <username>cn-north-1</username>
  <password></password>
</server>
<server>
  <id>z_mirrors</id>
</server>
```

pom.xml

```
<dependencies>
  <dependency>
    <groupId>a</groupId>
    <artifactId>a</artifactId>
    <version>a</version>
  </dependency>
</dependencies>
<distributionManagement>
  <repository>
    <id>release_cn-north-1</id>
    <url></url>
  </repository>
  <snapshotRepository>
    <id></id>
    <url></url>
  </snapshotRepository>
</distributionManagement>
</project>
```

2. Upload the local JAR package:

```
mvn deploy:deploy-file -DgroupId=com.huawei -DartifactId=aopalliance -Dversion=1.0 -Dpackage=jar
```

----End