CodeArts Artifact

FAQs

Issue 01

Date 2024-10-30

HUAWEI TECHNOLOGIES CO., LTD.

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 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 Repos?

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:

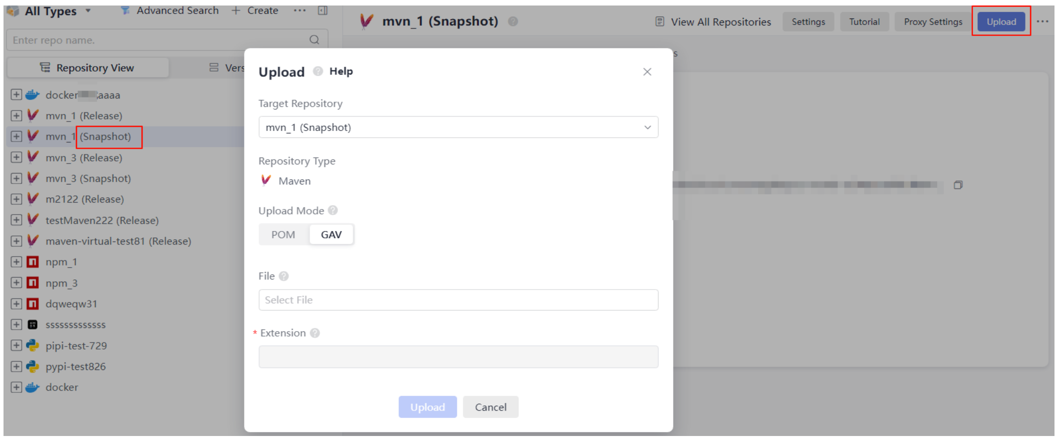
- Uploading Snapshots on the Release Repos Page
- Uploading Snapshots Using the Maven CLI
- Releasing Snapshots to a Maven Repository Through CodeArts Build

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**.



Target Repository: mvn_1 (Snapshot)
Repository Type: Maven
Upload Mode: POM [✓ GAV]
File: Select File
* Extension:

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.

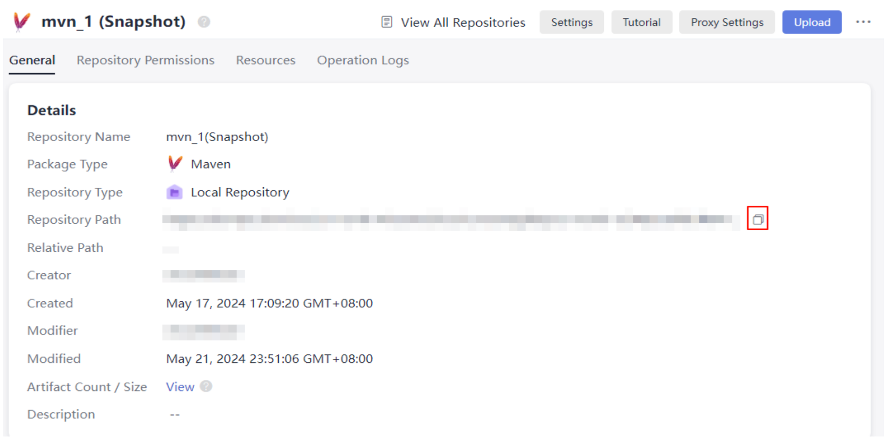
Step 4 Run **mvn deploy** to upload the Maven project.

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.



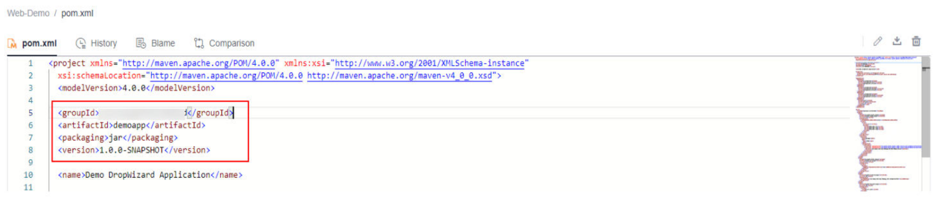
Details

Repository Name	mvn_1(Snapshot)
Package Type	Maven
Repository Type	Local Repository
Repository Path	[redacted]
Relative Path	
Creator	[redacted]
Created	May 17, 2024 17:09:20 GMT+08:00
Modifier	[redacted]
Modified	May 21, 2024 23:51:06 GMT+08:00
Artifact Count / Size	View
Description	--

----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.



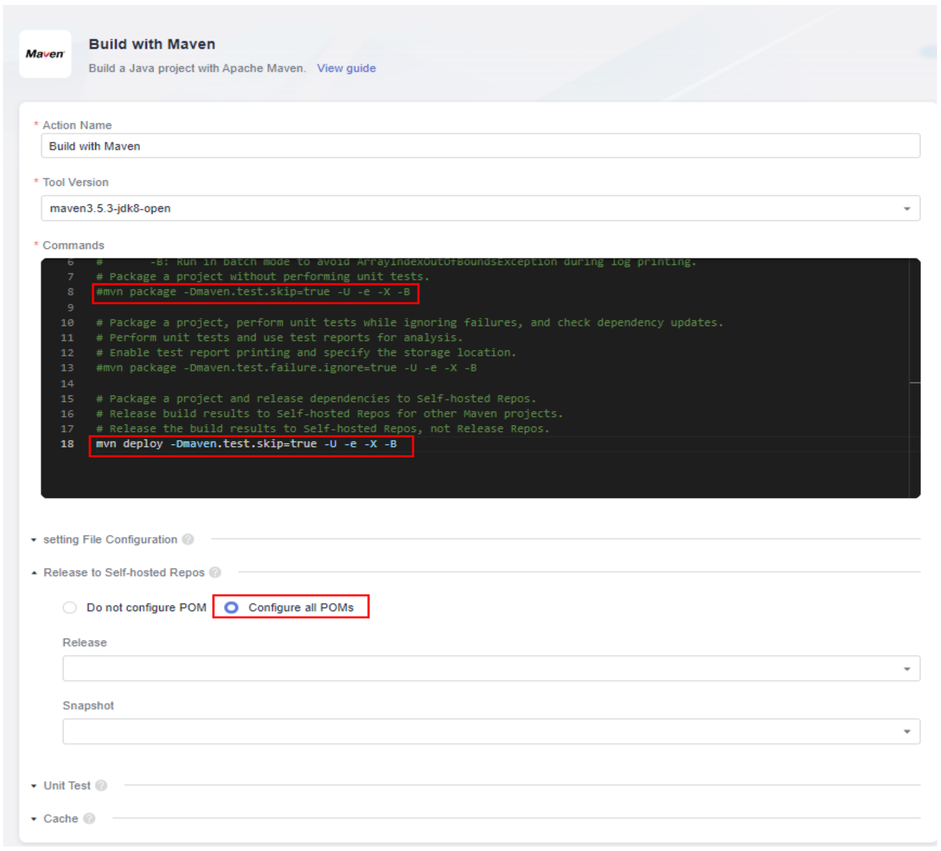
```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>[redacted]</groupId>
  <artifactId>demoapp</artifactId>
  <packaging>jar</packaging>
  <version>1.0.0-SNAPSHOT</version>

  <name>Demo DropWizard Application</name>

```

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**.



Action Name: Build with Maven

Tool Version: maven3.5.3-jdk8-open

Commands

```
#       -B: Run in batch mode to avoid ArrayIndexOutOfBoundsException during log printing.
# Package a project without performing unit tests.
#mvn package -Dmaven.test.skip=true -U -e -X -B

# Package a project, perform unit tests while ignoring failures, and check dependency updates.
# Perform unit tests and use test reports for analysis.
# Enable test report printing and specify the storage location.
#mvn package -Dmaven.test.failure.ignore=true -U -e -X -B

# Package a project and release dependencies to Self-hosted Repos.
# Release build results to Self-hosted Repos for other Maven projects.
# Release the build results to Self-hosted Repos, not Release Repos.
mvn deploy -Dmaven.test.skip=true -U -e -X -B
```

setting File Configuration

Release to Self-hosted Repos

Do not configure POM [✓ Configure all POMs]

Release:

Snapshot:

Unit Test

Cache

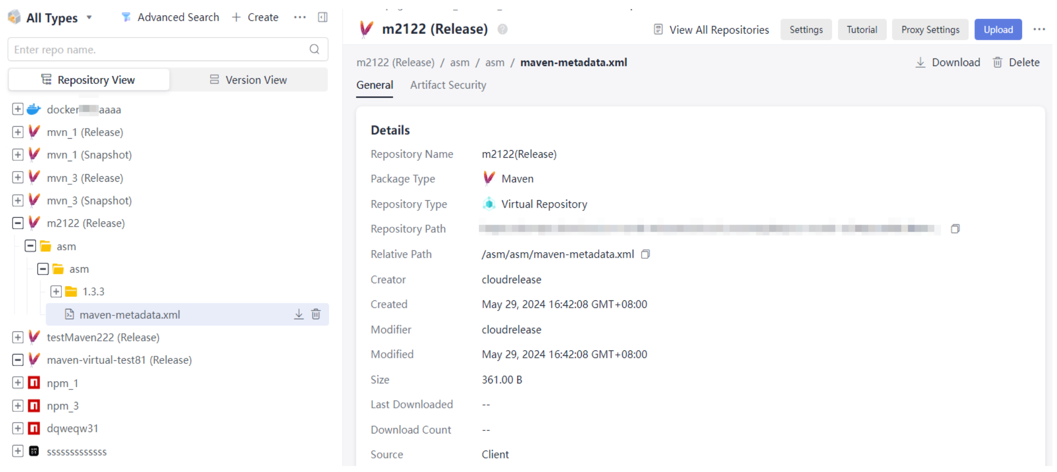
Step 3 Run a build task.

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.



Details

Repository Name	m2122(Release)
Package Type	Maven
Repository Type	Virtual Repository
Repository Path	[redacted]
Relative Path	/asm/asm/maven-metadata.xml
Creator	cloudrelease
Created	May 29, 2024 16:42:08 GMT+08:00
Modifier	cloudrelease
Modified	May 29, 2024 16:42:08 GMT+08:00
Size	361.00 B
Last Downloaded	--
Download Count	--
Source	Client

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?

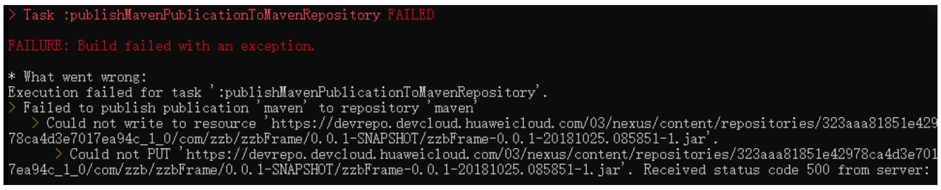
Yes.

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/323aaa81851e429
78ca4d3e7017ea94c_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/323aaa81851e42978ca4d3e701
7ea94c_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.

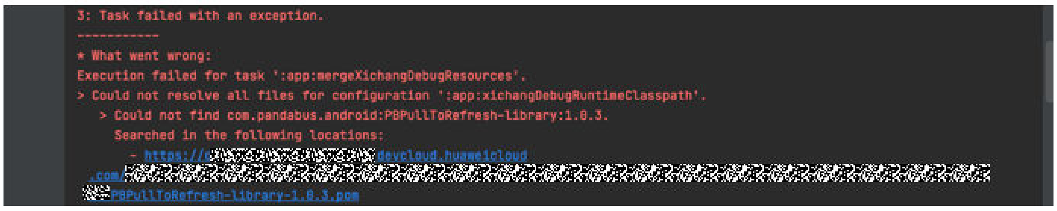
Solution

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

2.5 Why 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:1.0.3.
     Searched in the following locations:
       - https://[redacted]devcloud.huaweicloud
 .com/[redacted]
[redacted]PBPullToRefresh-library-1.0.3.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.

2.6 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:

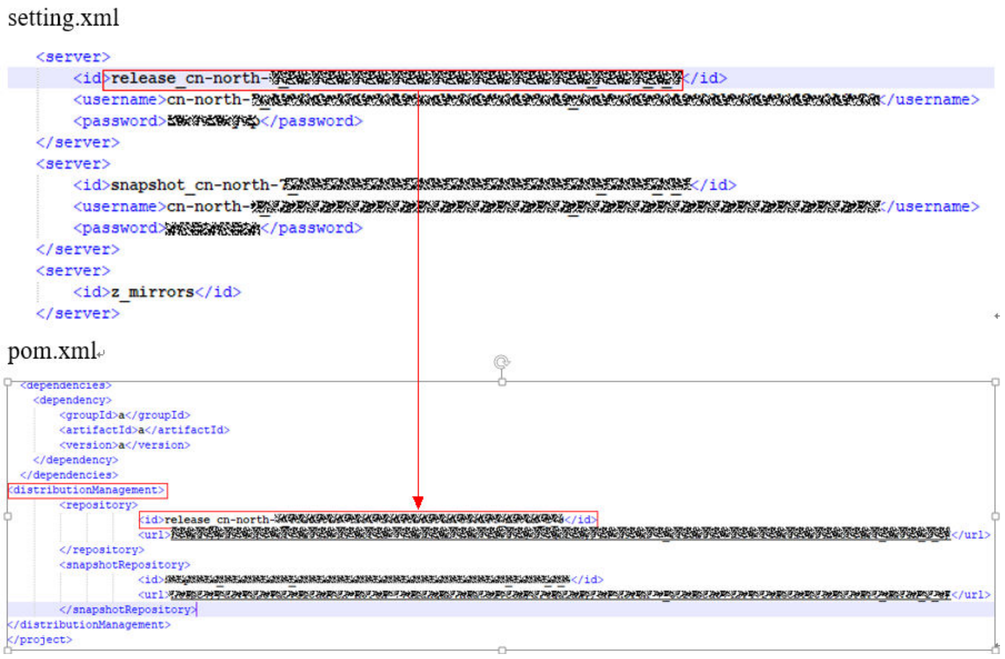
Step 1 Go to the self-hosted repo page, and choose Maven from the repository list.

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-[redacted]</id>
    <username>cn-north-[redacted]</username>
    <password>[redacted]</password>
</server>
<server>
    <id>snapshot_cn-north-[redacted]</id>
    <username>cn-north-[redacted]</username>
    <password>[redacted]</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-[redacted]</id>
            <url>[redacted]</url>
      </repository>
      <snapshotRepository>
            <id>[redacted]</id>
            <url>[redacted]</url>
      </snapshotRepository>
</distributionManagement>
</project>
```

2. Upload the local JAR package:

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

----End