CodeArts Artifact

FAQ

Issue 01

Date 2023-11-30

HUAWEI TECHNOLOGIES CO., LTD.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to "Vul. Response Process". For details about the policy, see the following website:https://www.huawei.com/en/psirt/vul-response-process For enterprise customers who need to obtain vulnerability information, visit: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 Repo Homepage Under CodeArts Artifact?

On the release repo homepage, the top-level directory names map to your project names. They are used to distinguish the project to which a software package belongs.

You can only browse files and directories in this directory and cannot upload files or create directories.

However, you can click a project name to access the project folder and upload files and create directories there.

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

No.

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

You can upload dependency packages to be referenced to Self-Hosted Repo.

1.3 Files in the Recycle Bin of a Release Repo Cannot Be Restored

Symptom

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

Cause Analysis

A file with the same name exists in the location to be restored 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: 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 the Self-hosted Maven Repo?

Background

Snapshots can be uploaded in the following modes:

- **Uploading Snapshots**
- **Uploading Snapshots Using the Maven CLI**
- **Releasing Snapshots to the Self-hosted Maven Repo Through CodeArts Build**

Uploading Snapshots

Step 1 Log in to CodeArts.

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

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

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 corresponding package.

----End

Uploading Snapshots Using the Maven CLI

Step 1 Access the **self-hosted Maven repo** homepage, and choose the snapshot repository in the repository list.

Step 2 Click **Set Me Up** in the upper right corner. The **Configuration File** dialog box is displayed.

.

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

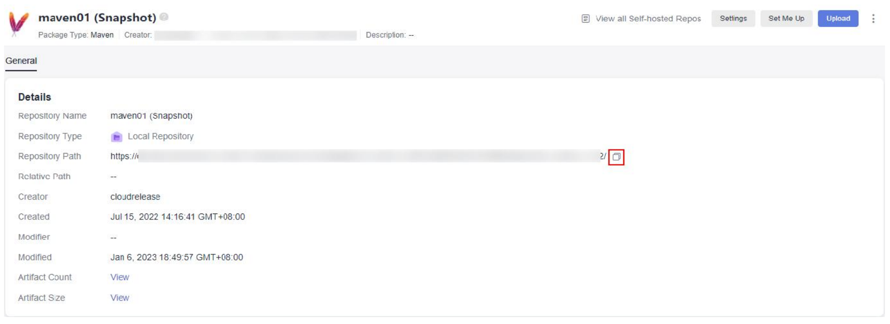
Step 4 Run **mvn deploy** to upload the Maven project.

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. -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 address, which can be obtained by clicking ❐ in the following figure.



Package Type: Maven | Creator: [redacted] | Description: --

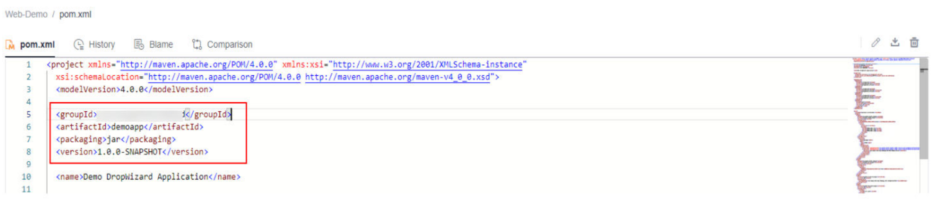
Details

Repository Name	maven01 (Snapshot)
Repository Type	Local Repository
Repository Path	https://[redacted]2/
Relative Path	--
Creator	cloudrelease
Created	Jul 15, 2022 14:16:41 GMT+08:00
Modifier	--
Modified	Jan 6, 2023 18:49:57 GMT+08:00
Artifact Count	View
Artifact Size	View

----End

Releasing Snapshots to the Self-hosted Maven Repo Through CodeArts Build

Step 1 Access the code repository, open the **pom.xml** file, and define the GAV information of the component to upload.



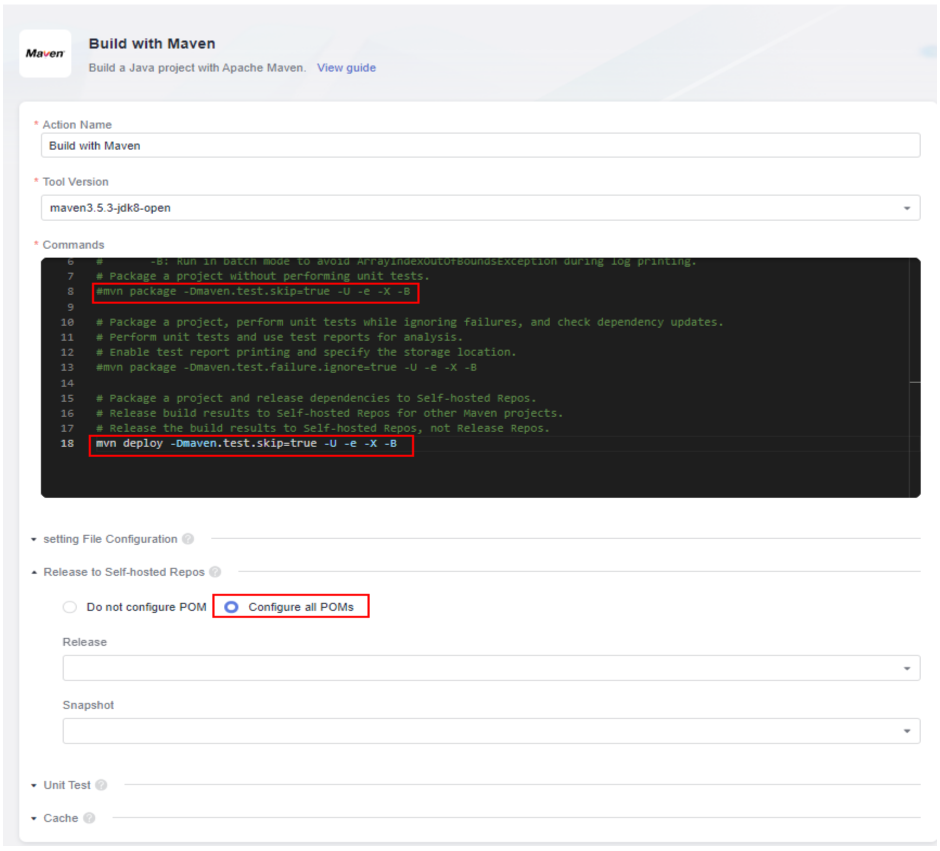
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <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 attributes uploaded to the self-hosted Maven repo 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. Specifically, 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 the build task. After the build task is executed, you can find the generated Maven component in the self-hosted Maven repo.

----End

2.2 How Do I Call a Private Component from a Self-hosted Maven Repo?

Step 1 Access the self-hosted repo, and click the name of a private component to open the file attribute page.

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

----End

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

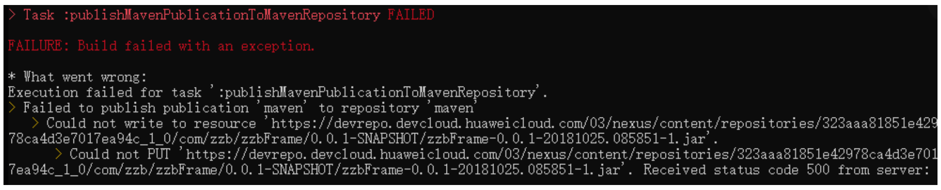
Yes.

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

2.4 Why Is Error Code 500 Returned When a Gradle Build Task Uploads a Maven Package?

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.

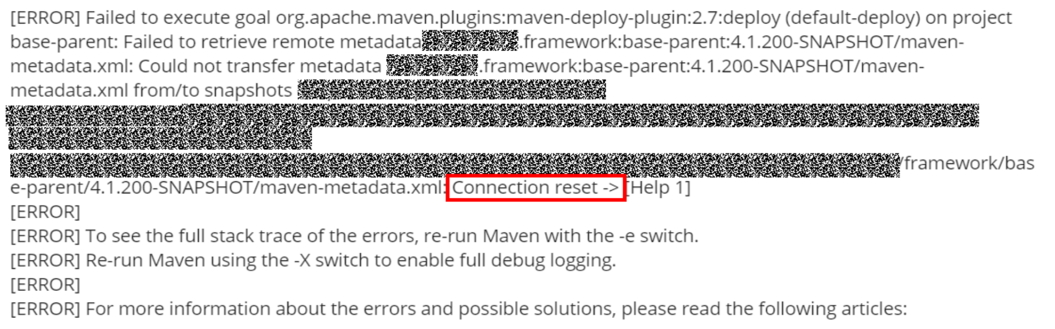
Solution

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

2.5 Why Can't the Repository Receive Requests?

Symptoms

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>
  ```

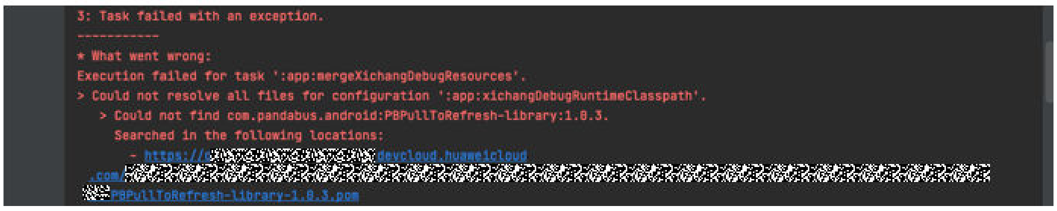
  You can also add the following command to your environment or build script.

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

Root Cause

The POM file is missing in the dependency.

When downloading dependency packages 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 code repository file **pom.xml** does not match the **settings.xml** file.

Solution

When uploading the 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 Access the self-hosted repo homepage, and choose Maven from the repository list.

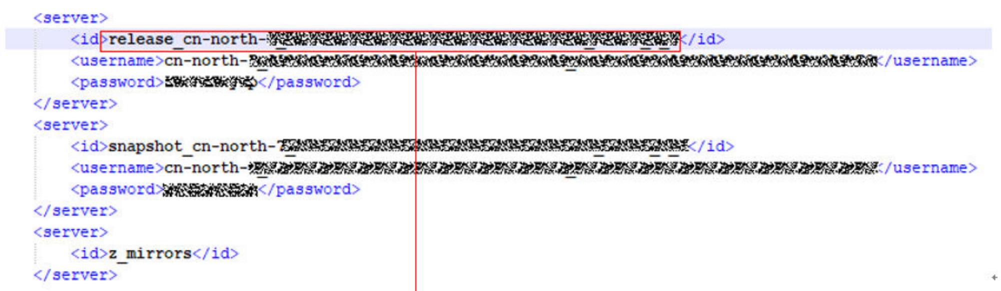
Step 2 Click **Set Me Up** in the upper right corner. The **Configuration Guide** dialog box is displayed.

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

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

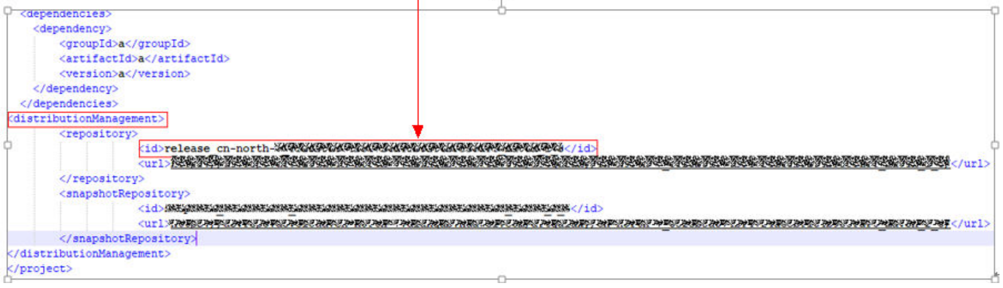
- In the Maven CLI, access the directory where the **pom.xml** file of the Maven project is stored, check whether the **repository_id** value in the **distributionManagement** element of the **pom.xml** file matches the **repository_id** value in the **server** element of the **settings.xml** file.

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>
```

- Upload the local JAR package:

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

----End