

制品仓库

# 常见问题

文档版本 03  
发布日期 2023-08-25



版权所有 © 华为技术有限公司 2023。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

---

# 目录

---

<b>1 软件发布库</b>	<b>1</b>
1.1 制品仓库服务下软件发布库首页为什么不能上传文件、创建目录等?	1
1.2 是否可以通过直接更改 pom.xml 文件中的依赖 ID 来调用软件发布库中的 jar 包?	1
1.3 软件发布库回收站中的文件无法还原	1
<b>2 私有依赖库</b>	<b>3</b>
2.1 如何将 snapshot 组件上传到 Maven 私有依赖库?	3
2.2 如何正确调用 Maven 私有依赖库的私有组件?	6
2.3 支持本地编译时调用私有依赖库中的软件包吗?	7
2.4 使用 gradle 构建任务上传 maven 包, 返回 500 错误提示	7
2.5 仓库接收不到请求 ( Connection Reset )	7
2.6 无法下载依赖的 war、jar 文件	8
2.7 本地构建 Maven 组件上传到私有依赖库, 返回 401 错误提示	8
2.8 客户端下载私有依赖库组件, 返回 http 错误码 403 提示	9

# 1 软件发布库

- 1.1 制品仓库服务下软件发布库首页为什么不能上传文件、创建目录等？
- 1.2 是否可以通过直接更改pom.xml文件中的依赖ID来调用软件发布库中的jar包？
- 1.3 软件发布库回收站中的文件无法还原

## 1.1 制品仓库服务下软件发布库首页为什么不能上传文件、创建目录等？

软件发布库页面中，列表中的顶层目录名对应租户下的项目名，用于区分归档软件包所属项目。

在该目录下不允许进行上传文件、创建目录等操作，只能浏览。

单击项目名称进入项目文件夹后即可进行上传文件、创建目录等操作。

## 1.2 是否可以通过直接更改 pom.xml 文件中的依赖 ID 来调用软件发布库中的 jar 包？

不可以。

因为发布库的软件包是用于部署，而不是用于编译构建过程中的依赖。

若需要引用依赖，请将依赖包上传至私有依赖库中。

## 1.3 软件发布库回收站中的文件无法还原

### 问题现象

在回收站页面上对文件执行还原操作，提示“存在重名文件”，无法还原。

### 原因分析

仓库中的被还原位置上存在一个同名的文件。

## 处理办法

可以选择“移动和替换”、“请勿移动”、“移动和重命名”三个操作。

移动和替换：从回收站还原的文件将会替换被还原位置的同名文件。

请勿移动：忽略回收站中该文件的还原操作。

移动和重命名：被还原位置将会同时保留原文件和回收站文件，从回收站还原的文件将被重命名。

# 2 私有依赖库

- [2.1 如何将snapshot组件上传到Maven私有依赖库?](#)
- [2.2 如何正确调用Maven私有依赖库的私有组件?](#)
- [2.3 支持本地编译时调用私有依赖库中的软件包吗?](#)
- [2.4 使用gradle构建任务上传maven包, 返回500错误提示](#)
- [2.5 仓库接收不到请求 \( Connection Reset \)](#)
- [2.6 无法下载依赖的war、jar文件](#)
- [2.7 本地构建Maven组件上传到私有依赖库, 返回401错误提示](#)
- [2.8 客户端下载私有依赖库组件, 返回http错误码403提示](#)

## 2.1 如何将 snapshot 组件上传到 Maven 私有依赖库?

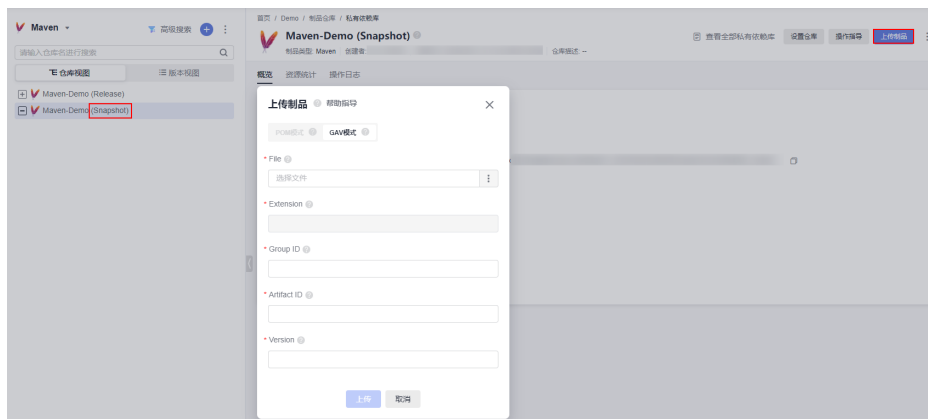
### 背景信息

制品仓库服务支持通过几种方式上传snapshot组件:

- [通过页面上上传snapshot组件](#)
- [通过Maven CLI命令上传snapshot组件](#)
- [通过编译构建将snapshot组件发布到Maven私有依赖库](#)

### 通过页面上上传 snapshot 组件

- 步骤1** 登录软件开发生产线首页。
- 步骤2** 单击导航“服务 > 制品仓库”，选择“私有依赖库”页签，找到对应的仓库。
- 步骤3** 在页面左侧仓库列表中选择snapshot仓库。单击“上传制品”，弹出“上传snapshot组件”对话框，请根据需要选择GAV定义来源。



GAV定义来源有以下两种：

GAV定义来源模式	描述
POM模式	从POM文件中解析出Maven的GAV坐标信息，然后再上传。
GAV模式	直接指定Maven包的坐标信息再上传。

**步骤4** 根据界面提示填写相关参数、上传对应文档完成组件上传。

----结束

## 通过 Maven CLI 命令上传 snapshot 组件

**步骤1** 进入“Maven私有依赖库”，在页面左侧导航中选择“snapshot”类型仓库。

**步骤2** 单击右上角“操作指导”弹出“私有依赖库使用配置”对话框。

。


**步骤3** 根据配置指导配置本地Maven工具。

**步骤4** 执行`mvn deploy`命令上传Maven项目。

在命令窗口进入maven项目“pom.xml”所在目录，输入如下命令，上传本地jar包：

```
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地址} -DrepositoryId=snapshots
```

## 📖 说明

- DgroupId、DartifactId、Dversion、Dpackaging：可根据需要自定义。
- Dfile：本地jar包所在位置的绝对路径。
- Durl：Maven Snapshot地址。单击图标即可获取。



----结束

## 通过编译构建将 snapshot 组件发布到 Maven 私有依赖库

**步骤1** 进入代码仓库，打开pom.xml文件，定义待上传组件的坐标信息（GAV信息）。



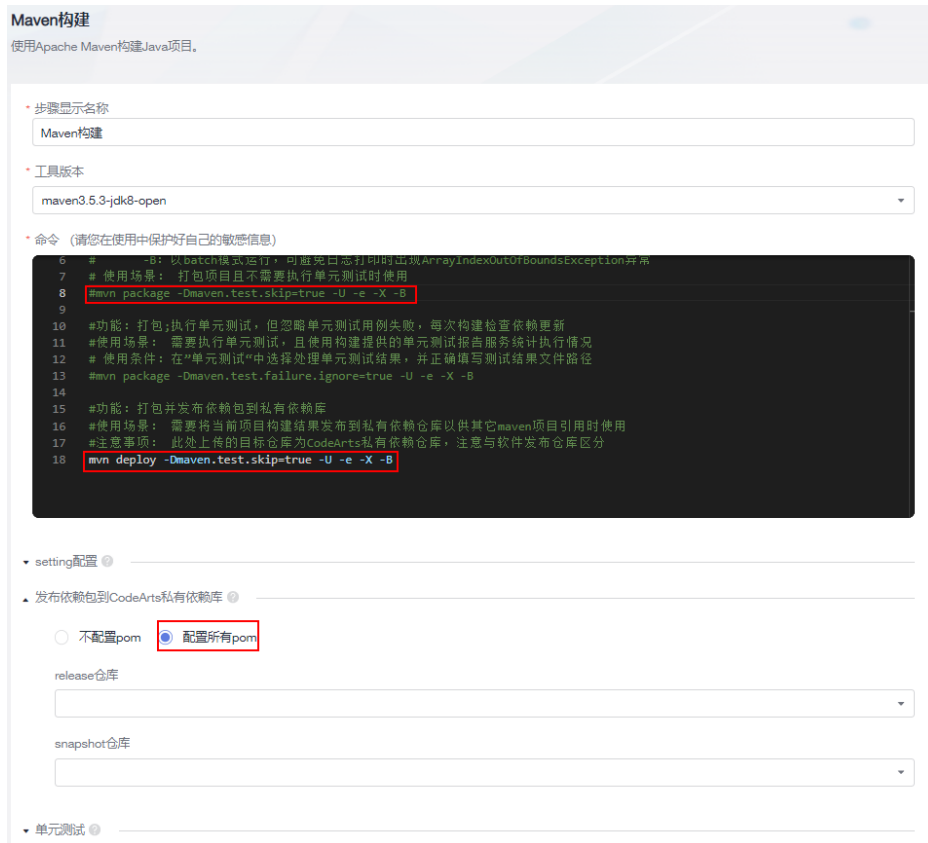
## 📖 说明

- 构建任务执行时会根据定义识别上传到私有依赖库的组件属性信息。
- version：版本号，默认上传release组件。如果需要上传snapshot组件，version值以“-SNAPSHOT”结尾即可，如“1.0-SNAPSHOT”。

**步骤2** 编辑编译构建任务，在“构建步骤Maven构建”中：

- 在“命令”框中：使用注释mvn package命令（在命令前加#），取消注释mvn deploy（删除其命令前的#）。
- 单击“发布依赖包到CodeArts私有依赖库”，勾选“配置所有pom”。



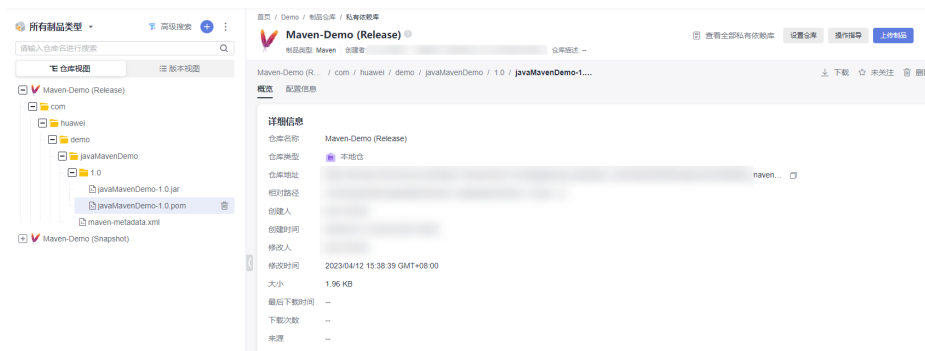


**步骤3** 执行构建任务，任务执行成功后即可在私有依赖库中找到生成的Maven组件。

----结束

## 2.2 如何正确调用 Maven 私有依赖库的私有组件？

**步骤1** 进入私有依赖库，单击需要私有组件名称，进入文件属性页面。



**步骤2** 获取依赖下载地址，复制后添加至“pom.xml”文件即可。

----结束

## 2.3 支持本地编译时调用私有依赖库中的软件包吗？

支持。

进入依赖包所在的仓库，单击页面右上角“操作指导”下载配置文件，参考弹框中的说明完成配置即可。

## 2.4 使用 gradle 构建任务上传 maven 包，返回 500 错误提示

### 问题现象

构建失败，日志信息提示如下：

```
> 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:
```

### 原因分析

由于snapshot地址错使用成了release地址导致。

### 处理方法

将地址改为snapshot地址，然后重新上传即可。

## 2.5 仓库接收不到请求（Connection Reset）

### 问题现象

本地构建失败，提示“Connection Reset”，日志信息提示如下：

```
[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 (https://devrepo.devcloud.cn-north-4.huaweicloud.com/[REDACTED].framework/base-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:
```

## 原因分析

用户使用的Java版本低，默认不支持TLS 1.2。

## 处理方法

- 若您使用的是Java 6版本：请升级到Java 8以上的版本。
- 若您使用的是Java 7版本：Java 7支持TLS 1.2，但在1.7.0\_131-b31之前的版本中默认禁用，可以使用以下命令启用。

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

或将以下命令添加到您的环境或构建脚本中。

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

## 2.6 无法下载依赖的 war、jar 文件

### 问题现象

本地工具通过私有依赖库无法下载组件，报错为找不到对应的pom文件，日志信息提示如下：

A terminal screenshot showing an error message. The text is as follows:

```
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://maven.aliyun.com/repository/public/com/pandabus/android/PBPullToRefresh-Library/1.8.3/pBPullToRefresh-Library-1.8.3.pom
```

### 原因分析

依赖中缺失对应的pom文件。

无论是通过gradle还是maven下载依赖，其顺序都为先下载pom文件，然后才是jar或者war，若云端不存在pom文件，则会下载失败。

### 处理方法

建议按照组件上传标准再次重新上传无法下载的组件。

## 2.7 本地构建 Maven 组件上传到私有依赖库，返回 401 错误提示

### 问题现象

通过本地IDE向私有依赖库上传Maven组件时报错：401-权限不足。

### 原因分析

代码仓库文件“pom.xml”中配置的私有依赖库信息，与私有依赖库配置文件“settings.xml”不符。

## 处理方法

上传组件时，将文件“pom.xml”中“distributionManagement”元素下“repository\_id”的值，替换为文件“settings.xml”中“server”元素下“repository\_id”的值。

上传组件操作步骤如下：

**步骤1** 进入“Maven私有依赖库”，在页面左侧导航中选择Maven类型仓库。

**步骤2** 单击页面右上角“操作指导”弹出“私有依赖库使用配置”对话框。

**步骤3** 根据配置指导配置本地Maven工具。

**步骤4** 执行mvn deploy命令上传Maven项目。

- 在命令窗口进入maven项目“pom.xml”所在目录，检查文件“pom.xml”中“distributionManagement”元素下的“repository\_id”的值是否与文件“settings.xml”中“server”元素下“repository\_id”的值是否匹配。

setting.xml

```
<server>
  <id>release_cn-north-1</id>
  <username>cn-north-1</username>
  <password>cn-north-1</password>
</server>
<server>
  <id>snapshot_cn-north-1</id>
  <username>cn-north-1</username>
  <password>cn-north-1</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>http://192.168.1.100:8081</url>
  </repository>
  <snapshotRepository>
    <id>snapshot_cn-north-1</id>
    <url>http://192.168.1.100:8081</url>
  </snapshotRepository>
</distributionManagement>
</project>
```

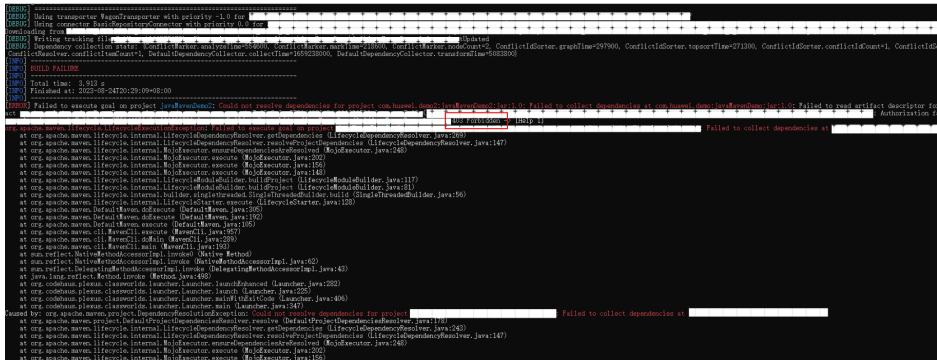
- 执行如下命令，上传本地jar包：  
mvn deploy:deploy-file -DgroupId=com.huawei -DartifactId=aopalliance -Dversion=1.0 -Dpackagi=jar

----结束

## 2.8 客户端下载私有依赖库组件，返回 http 错误码 403 提示

### 问题现象

从客户端下载私有依赖库组件失败，提示http错误码403。



## 原因分析

- 1. 当前用户无对应私有依赖库的权限。
- 2. 可能是租户当月使用的制品仓库服务流量大于当前已购租户套餐的规格。

## 处理方法

- 1. 联系项目管理员添加对应的权限。
- 2. 检查租户制品仓库服务的下载流量是否耗尽，若耗尽且还需要继续使用下载功能，请升级套餐。