

制品仓库

常见问题

文档版本 01
发布日期 2024-10-18



版权所有 © 华为技术有限公司 2024。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。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

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

如企业客户须获取漏洞信息，请参见如下网址：

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

目录

1 软件发布库	1
1.1 在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该如何处理?	1
2 私有依赖库	3
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?	5
2.3 支持本地编译时调用私有依赖库中的软件包吗?	6
2.4 使用 gradle 构建任务上传 maven 包, 返回 500 错误提示, 该如何处理?	6
2.5 仓库接收不到请求 (Connection Reset) 该如何处理?	6
2.6 无法下载依赖的 war、jar 文件时怎么办?	7
2.7 本地构建 Maven 组件上传到私有依赖库, 返回 401 错误提示, 该如何处理?	8

1 软件发布库

1.1 在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。

如果需要引用依赖，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该如何处理？

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？

背景信息

制品仓库服务支持通过以下方式上传snapshot组件：

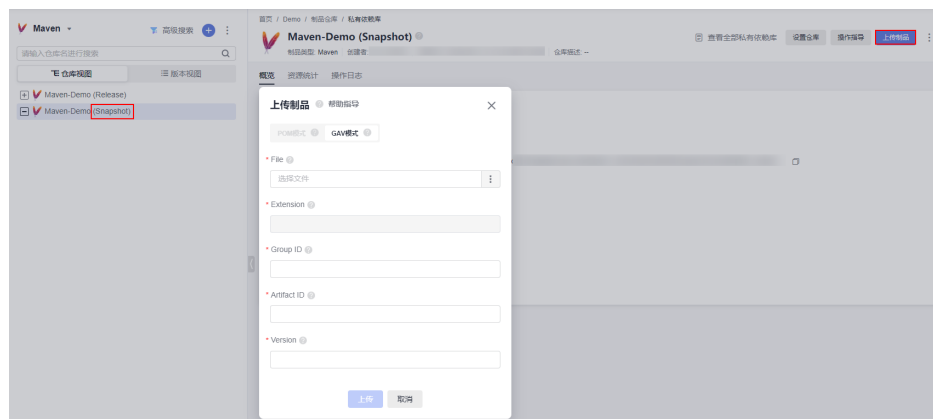
- [通过页面上传snapshot组件](#)
- [通过Maven CLI命令上传snapshot组件](#)
- [通过编译构建将snapshot组件发布到Maven私有依赖库](#)

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“操作指导”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）。

```
pom.xml 大小: 3.88KB 全屏显示 复制源代码 下载 编辑 移除 修改追踪
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5   <groupId>com.huawei</groupId>
6   <artifactId>demoapp</artifactId>
7   <packaging>jar</packaging>
8   <version>1.0.0-SNAPSHOT</version>
9
10  <name>Demo Drowizard Application</name>
```


📖 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 在“构建步骤Maven构建”中，编辑编译构建任务：

- 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”。

Maven构建
使用Apache Maven构建Java项目。

• 步骤显示名称
Maven构建

• 工具版本
maven3.5.3-jdk8-open

• 命令 (请您在使用中保护好您的敏感信息)

```
0 # -B: 以batch模式运行, 可避免日志打印时出现ArrayIndexOutOfBoundsException异常  
1 # 使用场景: 打包项目且不需要执行单元测试时使用  
2 # mvn package -Dmaven.test.skip=true -U -e -X -B  
3  
4 #功能: 打包;执行单元测试, 但忽略单元测试用例失败, 每次构建检查依赖更新  
5 #使用场景: 需要执行单元测试, 且使用构建提供的单元测试报告服务统计执行情况  
6 # 使用条件: 在“单元测试”中选择处理单元测试结果, 并正确填写测试结果文件路径  
7 #mvn package -Dmaven.test.failure.ignore=true -U -e -X -B  
8  
9  
10 #功能: 打包并发布依赖包到私有依赖库  
11 #使用场景: 需要将当前项目构建结果发布到私有依赖库以供其它maven项目引用时使用  
12 #注意事项: 此处上传的目标仓库为CodeArts私有依赖库, 注意与软件发布仓库区分  
13 # mvn deploy -Dmaven.test.skip=true -U -e -X -B
```

• setting配置

• 发布依赖包到CodeArts私有依赖库

不配置pom 配置所有pom

release仓库

snapshot仓库

• 单元测试

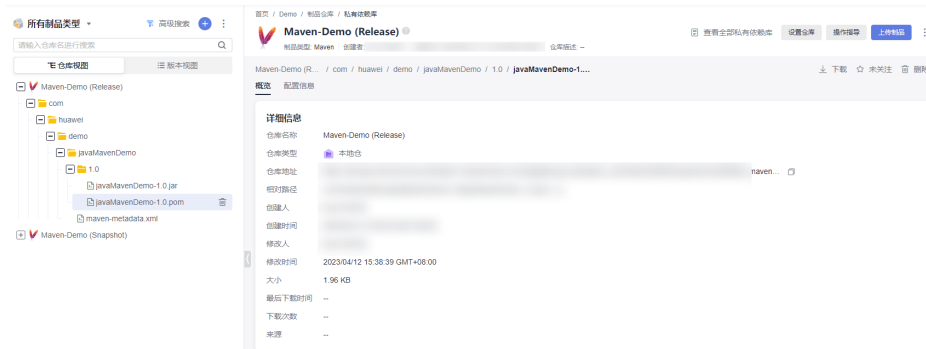
步骤3 执行构建任务。

待任务执行成功后，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，单击需要调用的私有组件名称，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，该如何处理？

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 ）该如何处理？

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]
[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:
```

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>
```

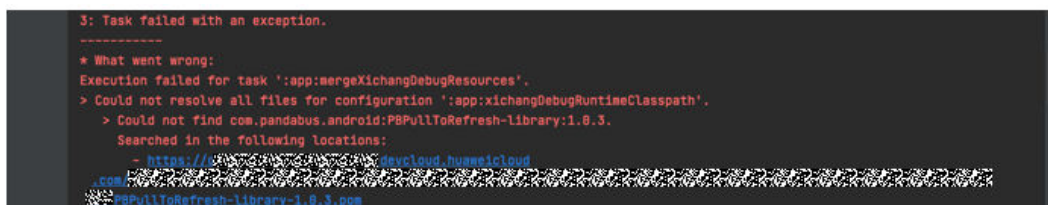
或将以下命令添加到您的环境或构建脚本中。

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

2.6 无法下载依赖的 war、jar 文件时怎么办？

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：



```
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/[REDACTED]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，该如何处理？

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。

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

2. 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