Mule Project Offline Packaging

This document serves as how to guide to perform Mule project offline packaging. This guide is for packaging environment, which is airtight without any connection to the internet, and do not have implementation of local maven repositories servers such as Nexus.

For example, in some banking or government organization, they require code packaging to be done internally by the internal deployment team in secure air-tight environment where no internet access is allowed. Hence you will not be able to connect to the external maven repositories to download the dependencies that are needed to package your Mule project

Pre-Requisite

1. Developer workstation (able to access the internet).
2. Credential's setup to connect to Mule Enterprise Repositories (Refer here for the steps)
3. Sample Mule Project to be packaged

Step-by-step guide

In your local machine

1. All the dependencies that are needed by your project will be downloaded to your machine maven repository. The default location of your local maven repository will be in your user folder, under .m2/ repository. (e.g., /Users/**<userName>**/.m2/repository)

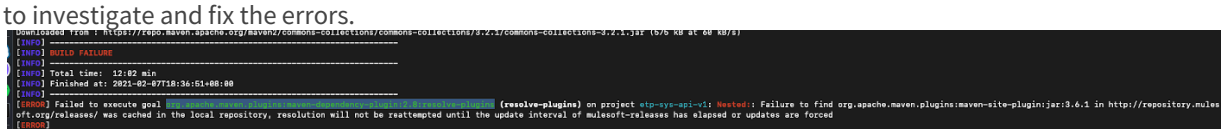
Operating System	Default Maven Location	Default Maven Local Repository
Windows	C:\Users\<user_name>\.m2	C: \Users\<user_name>\.m2\repository
Linux	/home/<user_name>/.m2	/home/<user_name>/.m2/repository
Mac	/Users/<user_name>/.m2	/Users/<user_name>/.m2/repository

2. Edit the settings.xml to set the local repository to another location, this is to ensure that, only dependencies that are needed are being transferred to the air-tight packaging/deployment machine/vm.
3. Add the following *<localRepository>**<temp location>***</localRepository>*to your settings.xml as shown below. Refer here for more information on the ***localRepository*** field.

```
<settings>
 <localRepository>/Users/tkianhock/Documents/tempFolder</localRepository>
    <servers>
```

4. In your **<mule project home>** directory, run ***mvn dependency:go-offline***, this command will package your mule project and download all the dependencies that are needed by your mule project to the ***<temp location>*** folder***.*** If there is any authentication or authorization error, ensure your credentials to the Mule Enterprise Repository and to the Exchange are configured correctly as per prerequisite no 2. You can proceed to step 5 if you encounter "resolve-plugin goal" issue as shown below, for other errors you will need

to investigate and fix the errors.



5. After all the dependencies have been downloaded, execute ***mvn clean package*** to download any additional dependencies needed to package your project. If this step is successful, all the dependencies that is needed to package your project are available in ***<temp location>*** folder**.**
6. To test offline maven packaging within your local machine, disconnect your machine from the internet.
7. Before executing the following steps, make a backup of the settings.xml by running the following command ***cp <default maven location>/settings.xml <default maven location>/settings-online.xml.***
8. Set the ***updatePolicy flag*** to false in maven settings.xml. The ***updatePolicy flag*** can be found under repository/repositories/releases/ and repository/repositories/snapshots/. Refer here for more information on the ***updatePolicy*** flag

```
<repository>
    <id>nexus-ee</id>
    <name>nexus-ee</name>
    <url>https://repository.mulesoft.org/nexus-ee/content/repositories/releases-ee/</url>
    <layout>default</layout>
    <releases>
        <updatePolicy>never</updatePolicy>
        <enabled>true</enabled>
    </releases>
    <snapshots>
        <enabled>true</enabled>
        <updatePolicy>never</updatePolicy>
    </snapshots>
</repository>
```

9. If your project is using parent pom, set the reference of the parent pom location to a physical path rather than relative path
10. Execute ***mvn -nsu -o clean package***. Your mule project will be packaged without internet connectivity in your local machine.

In the target machine/vm (air-tight environment)

1. Install maven in the target machine/VM. Refer here for information on maven installation. This step needs to be executed once.
2. Copy settings.xml file from your local machine to the target machine/vm. Remove the <localRepository>**<temp location>**</localRepository> from settings.xml so that maven will use the default location to search for the dependencies needed to build your project. This step only needs to be executed once in the target machine/vm.
3. Copy all the folders under ***<temp location>*** in your local machine to the target machine/vm default maven local repository location.

Operating System	Default Maven Location	Default Maven Local Repository
Windows	C:\Users\<user_name>\.m2	C:\Users\<user_name>\.m2\repository
Linux	/home/<user_name>/.m2	/home/<user_name>/.m2/repository

Operating System	Default Maven Location	Default Maven Local Repository
Mac	/Users/<user_name>/.m2	/Users/<user_name>/.m2/repository

4. Copy the mule project folder to the air-tight packaging/deployment machine/vm without the target folder. Alternatively, you also can check in the Mule project your company/client internal source code repository which is accessible by the target machine/vm. Perform a git clone or git pull in the target machine/vm to download the latest copy of your source codes
5. Navigate to your mule project location and run ***mvn -nsu -o clean package*** to package your Mule project. The packaged Mule Project can be handed over to the deployment team to deploy to the respective Mule Runtimes. You can also build the Jenkins pipeline to automate step 4 , 5 and deployment of the package to your targeted Mule Environment.
6. Step 4 and 5 will be reiterative task, where you can continue to developed & build your Mule packages to test it in your targeted Mule Runtime environment (assuming there are no new dependencies being added to your project).

In your local machine (Restoring back the original settings.xml)

1. You can change back your settings.xml to the original version by making backup of the current settings.xml by invoking ***cp <default maven location>/settings.xml <default maven location>/settings-offline.xml***
    a. Rename the setting-online.xml to settings.xml by executing the following command ***mv <default maven location>/settings-online.xml <default maven location>/settings.xml.***
2. Remove or comment out the line ***<localRepository><temp location></localRepository>*** in your settings.xml to point your maven to the default maven repository location.

Additional dependencies needed for your project

Perform the following steps for additional dependencies that is needed by your project

1. Repeat the steps in the ***"In your local machine"*** section.
2. Repeat steps 3-5 in the ***"In the target machine/vm (air-tight environment)"*** section
3. Repeat the steps in the ***"In your local machine (Restoring back the original settings.xml)"*** section.