

Title: Multi Module Maven Project and Docker: Cannot find artifact?

Post Body:

I have the following multi-stage dockerfile that attempts to copy each module of my multi-module java maven project and build my ear file using mvn clean install.

**dockerfile:**

```
# Copy files from local to maven image and build ear FROM maven:3.5-jdk-8 AS build COPY module1 /usr/src/app/src COPY module2 /usr/src/app/src
```

The error that I am getting is as follows:

```
Step 8/20 : RUN mvn -f /usr/src/app/pom.xml clean install ---> Running in cf9d8c1ef9ed [INFO] Scanning for projects... [ERROR] [ERROR] Could not find artifact com.company.parent:jee6:1.0.1-SNAPSHOT
```

The section in the pom.xml that corresponds to this error:

```
<parent> <groupId>com.company.parent</groupId> <artifactId>jee6</artifactId> <version>1.0.1-SNAPSHOT</version> </parent>
```

I assume this issue because when trying to run the command in the maven docker image it cannot see my local .m2 folder? And the dependency that it is looking for is a private dependency on my local machine.

Would also copying my maven settings.xml help?

How can I resolve this? I do not want someone to have to have Maven installed on their machine in order to run this dockerfile.

Accepted Answer:

Dependency com.company.parent:jee6:pom:1.0.1-SNAPSHOT seems to be private, your Maven command inside Docker build needs to be able to either download it from private repository or have it readily accessible.

I assume this issue because when trying to run the command in the maven docker image it cannot see my local .m2 folder?

Yes, it then cannot see your settings.xml with private repository config, or local dependency if it's already available locally.

Would also copying my maven settings.xml help?

It's better not to: your settings.xml (and eventual secrets within) may be available to anyone using your image later. Using a secret mount with BuildKit would be a better solution (see below)

---

You have multiple solutions:

## Mount settings.xml as secret during build

This solution assumes you have a settings.xml [configured with proper credentials to access private registry](#).

Use [Docker BuildKit](#) with `--mount=secret` to load settings.xml as secret with a Dockerfile such as:

```
# syntax=docker/dockerfile:1.2 # Required comment at top of Dockerfile for using BuildKit FROM maven:3.5-jdk-8 AS build COPY parent-pom.xml /tmp/parent/pom.xml
```

And build command such as:

```
DOCKER_BUILDKIT=1 docker build --secret id=mvnsettings,src=$HOME/.m2/settings.xml .
```

Maven should then be able to download parent dependency during build.

*Note: this is NOT COPYING the settings.xml in image, as the secret settings.xml will only be made available for the specified build step and won't be persisted in final image.*

## Copy com.company.parent:jee6 pom.xml during build

This solution is less practical and may not solve problem entirely:

- It would require to have com.company.parent:jee6:pom:1.0.1-SNAPSHOT pom.xml file available in build context
- Your parent pom.xml may refer to other private dependencies. You would have to include them the same way.

... But it still may be worth a try.

You can do something like:

```
FROM maven:3.5-jdk-8 AS build # Copy and install parent pom COPY parent-pom.xml /tmp/parent/pom.xml RUN mvn -f /tmp/parent/pom.xml clean install
```

Highest Rated Answer:

There are 2 different parts to the error: Cannot find the parent, and the parent.relativePath element is wrong. I think the 2nd part might be causing the 1st.

Since your parent element doesn't specify a relativePath element, the default is the module parent path (aka ..). Your modules are not in a child of the parent folder (/usr/src/app) but rather in the (/usr/src/app/src subfolder).

Try changing your copy commands to :

```
COPY module1 /usr/src/app COPY module2 /usr/src/app  COPY module3 /usr/src/app  COPY pom.xml /usr/src/app
```

You should then see:

```
/usr/src/app /usr/src/app/module1 /usr/src/app/module2 /usr/src/app/module3
```