

Title: Non-resolvable parent POM on Docker Build

Post Body:

I'm trying to build a Docker image from my DockerFile but keep getting an error like it can't find the parent `pom.xml` to perform a maven command in the docker file and build the project. I've been looking around and you see what people do is the add to the child `pom.xml` a reference to the parent `pom.xml` y tried adding a `relativePath>.. /pom.xml/relativePath>` to the child but still won't work.

## Maven-multimodule project

[

## DockerFile

```
FROM alpine/git as clone WORKDIR /app RUN git clone https://github.com/RicardoVargasLeslie/manager.git FROM openjdk:8-jdk-alpine
```

## Child-pom.xml(Web)

```
<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://www.w3.org/2001/XMLSchema.xsd'>
```

## Parent-pom.xml(Core)

```
<?xml version='1.0' encoding='UTF-8'?> <project xmlns='http://maven.apache.org/POM/4.0.0' xmlns:xsi='http://www.w3.org/2001/XMLSchema.xsd'>
```

## Docker comand to build image

```
docker build -t rest-api .
```

## Comand-line Trace

```
$ docker build -t rest-api . Sending build context to Docker daemon 42.92MB Step 1/11 : FROM alpine/git as clone ----> ald22e2b1c0d
```

I'm not sure what wron or how to make the build work,thanks for any help.

Accepted Answer:

Maven is a tool that enables a declarative way of managing your project's dependencies. It does that by standardising the way jar and pom files (artifacts) are shared by projects and uses maven coordinates to declare a project dependency (groupId, artifactId and version) which are then looked up in a large public maven repository of jar and pom files called Maven Central.

In order to have your jar/pom files (like your core for example) managed as a maven artifact, you need to deploy your artifact to either the Maven Central (public repository accessible by anyone) or a private repository some where in your private network (or even on the internet, and maybe with authentication). In the case of a private repository, you need to include the you private repository so that your maven execution not only looks for artifacts in Maven Central, but also in your private repository. You can achieve this by adding the following to your `settings.xml` file:

```
<!-- Authentication here --> <servers> <server> <id>nexus</id> <username>some_username</username> <password>some_password</password>
```

I noticed your child `pom.xml` file references a custom parent module. You need to make sure that your parent module is in a maven repository (Nexus/Artifactory) and that you supply a `settings.xml` file with the repository settings for the Docker container to be able to pull the parent pom.

Highest Rated Answer:

In my case i was getting this due to the fact that apparently the Dockerfile instructions as copy can not see the parent folders !

meaning if you have a structure :

```
ParentProject | --> Child Module A --> pom.xml --> DockerFile | --> Child Module B --> pom.xml --> DockerFile | -> pom.xml
```

the Dockerfiles inside the modules can't perform actions like :

```
COPY ../pom.xml /
```

because it can't access localSystem ".." !! the solution was to move the Dockerfiles to the parent folder like this :

```
ParentProject | --> Child Module A --> pom.xml | --> Child Module B --> pom.xml | --> DockerFileA --> DockerFileB -> pom.xml
```

and then i was just doing something like this :

```
WORKDIR /app COPY ./pom.xml /app COPY ./ChildA/pom.xml /app/child-project/
```

It took me tons of times to figure this out, i needed to create a container just with COPY instructions without running maven , and then i drilled into the container files and found that the pom parent was not copied and actually somehow the docker copied the child pom as the parent pom for this instruction :

```
COPY ../pom.xml /
```

so you should know that when building docker , you can't copy files from parent folders in your localMachine since it does not have access to it. and it will execute the same command on the current folder, which makes it really hard to detect since no error is thrown upon it scenario.