

Maven -- How to

ALL HOW TOs

- [Generate jar/war](#)
 - [Reload all dependencies in pom.xml](#)
 - [Run all steps of Maven](#)
 - [Open an SVN project as a Maven project in Eclipse](#)
 - [Creating a local repository holding jars not available in a public repository](#)
-

Generate jar/war

- Edit pom.xml
 - Set Packaging to either jar or war
 - Save pom.xml
 - Rt click pom.xml
 - Select Run as... -> Maven Install
 - Look in target directory for *.jar or *.war
-

Reload all dependencies in pom.xml

- Rt click pom.xml -> Run as -> Maven install
-

Run all steps of Maven

- Rt click pom.xml -> Run as -> Maven build
 - Select package (May need to select something else. Not exactly sure why package is the right thing to select. Got this info from Matt.)
-

Open an SVN project as a Maven project in Eclipse

In Windows Explorer

- Navigate to Eclipse working directory
- Rt Click -> SVN Checkout...
- Use ... browse button for URL of repository to select project to checkout.
- Click OK

In Eclipse

- File -> Import...
 - Select Maven -> Existing Maven Projects
 - Browse... to project in Eclipse working directory
 - select trunk
 - OK
 - Finish
-

Creating a local repository holding jars not available in a public repository

Assumptions

- The jar was created by Maven in another project with the following

```
<groupId>edu.cornell.clo</groupId>
<artifactId>messageQueue</artifactId>
<version>0.4.1</version>
<packaging>jar</packaging>
```

In Project (that wants to access the jars)

- create repo directory just off the project base directory
- for each jar to be accessed locally
 - add directories for each level of the groupId (ex. ./repo

)

- add jar name (aka artifactId) without the version (ex. /repo/edu/cornell/clo
-)
- add directory for the version of the jar (ex. /repo/edu/cornell/clo/messageQueue
-)
- put the jar in that directory (ex. /repo/edu/cornell/clo/messageQueue/0.4.1
-)

In POM (for the project that wants to access the jars)

- define the local repository

```
<repositories>
  <repository>
    <id>data-local</id>
    <name>data</name>
    <url>file://${project.basedir}/repo</url>
  </repository>
</repositories>
```

- add the dependency on the local jar

```
<dependency>
  <groupId>_package-path_</groupId>
  <artifactId>_jar-name-without-version_</artifactId>
  <version>_version_</version>
</dependency>
```

From our example above, this would be...

```
<dependency>
  <groupId>edu.cornell.clo</groupId>
  <artifactId>messageQueue</artifactId>
  <version>0.4.1</version>
</dependency>
```

Rebuild

- Follow instructions above for [Run all steps of Maven](#)
-