

# Hibernate Tip: Add Metamodel classes to project

## Question:

I use Hibernate's Static Metamodel Generator to generate the JPA Metamodel. These classes are generated to a different directory which isn't used as a source folder. Is there any way to automatically register this folder as a source folder?

## Solution:

This is a question I asked myself and the reviewers of my book. As I learned, there is a Maven plugin which can do exactly that. Special thanks to Frits Walraven who's reviewing the book and showed me the plugin.

The only thing you need to do is to add the following Maven plugin to your build configuration. It registers a list of directories as additional source folders. I use it in the parent pom file of my project to add the directory, to which the [JPA Metamodel](#) classes get generated *target/generated-sources/annotations*, as a source folder.

```
<project>
...

<build>
  <plugins>
    <plugin>
      <groupId>org.codehaus.mojo</groupId>
      <artifactId>build-helper-maven-plugin</artifactId>
      <version>3.0.0</version>
```

# Hibernate Tip: Add Metamodel classes to project

```
<executions>
    <execution>
        <id>add-source</id>
        <phase>generate-sources</phase>
        <goals>
            <goal>add-source</goal>
        </goals>
        <configuration>
            <sources>
                <source>target/generated-
sources/annotations</source>
            </sources>
        </configuration>
    </execution>
</executions>
</plugin>
</plugins>
</build>

...
</project>
```

# Hibernate Tip: Add Metamodel classes to project

## Source Code

Get the source code from [GitHub](#)

## Learn more:

The JPA Metamodel provides a type-safe way to reference entity attributes when you create a *CriteriaQuery* or an *EntityGraph*.

I explain it in more detail in: [Create type-safe queries with the JPA static metamodel](#).