

## Usage

The `source` plugin can be used to create a `jar` file of the `project sources` from the command line or by binding the `goal` to the `project's build lifecycle`. To generate the `jar` from the command line, use the following command:

```
mvn source:jar
```

A `jar` file of the `test sources` can also be generated by executing:

```
mvn source:test-jar
```

### Installing the sources along with your artifact

There are two ways to do this. You can either bind this plugin to a phase or you can add it to a profile. The `goals` `source:jar-no-fork` and `source:test-jar-no-fork` are preferred for binding the `goal` to the `build lifecycle`.

### Installing the sources using a phase binding

Here is how you would configure the plugin in your `pom.xml` to run automatically during the `verify` phase:

```
<project>
...
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-source-plugin</artifactId>
      <version>2.4</version>
      <executions>
        <execution>
          <id>attach-sources</id>
          <phase>verify</phase>
          <goals>
            <goal>jar-no-fork</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
...
</project>
```

We are using the `verify` phase here because it is the phase that comes before the `install` phase, thus making sure that the `sources jar` has been created **before** the `install` takes place.

### Installing the sources using a profile

If you want to install a `jar` of your sources along with your artifact during the release process, you can add this to your `pom.xml` file:

```
<project>
...
<profiles>
  <profile>
    <id>release</id>
```

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-source-plugin</artifactId>
      <version>2.4</version>
      <executions>
        <execution>
          <id>attach-sources</id>
          <goals>
            <goal>jar-no-fork</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
</profile>
</profiles>
...
</project>
```