

# Migrating from org.sonatype.aether to Apache Maven Resolver

Much existing, legacy Maven plugin code uses org.sonatype.aether to interact with (download, upload, query) Maven repositories such as the central repository.

Some plugin use this for tests, especially integration tests, but not in the plugin itself.

This page describes how to convert that old code to the current APIs from Maven resolver.

## Artifacts, Libraries, and Dependencies

The oldest dependencies have the group ID org.sonatype.aether.

Old dependencies have the group ID org.eclipse.aether.

The up to date code lives in org.apache.maven.resolver.

To be specific:

org.sonatype.aether:aether-connector-file org.apache.maven.resolver:maven-resolver-transport-file

org.sonatype.aether:aether-connector-wagon org.apache.maven.resolver:maven-resolver-transport-wagon

When migration is complete, no pom.xml should import a dependency with the group ID org.sonatype.aether or org.eclipse.aether

Web searches will uncover a lot of outdated, buggy documentation based on org.sonatype.aether and org.eclipse.aether. In particular ignore all documentation on eclipse.org.

## Packages

Old code lives in the org.sonatype.aether package. When you're done no references to this package should remain.

New code lives in the org.eclipse.aether package. This includes code that comes from org.apache.maven.resolver. That is, in the newest code the group ID does not match the Java package name.

You cannot simply rename packages in the Java source code. Subpackages have also been renamed or moved and classes have been renamed or deleted as well.

## Step-by-step guide

1. Add a dependency on Apache Maven resolver to the necessary modules. In most cases these will be modules that already depend on org.sonatype.aether. For example, if you already have

```
<dependency>
  <groupId>org.sonatype.aether</groupId>
  <artifactId>aether-connector-file</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.sonatype.aether</groupId>
  <artifactId>aether-connector-wagon</artifactId>
  <scope>test</scope>
</dependency>
```

You would add

```
<dependency>
  <groupId>org.apache.maven.resolver</groupId>
  <artifactId>maven-resolver-transport-file</artifactId>
  <version>1.4.1</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.apache.maven.resolver</groupId>
  <artifactId>maven-resolver-transport-wagon</artifactId>
  <version>1.4.1</version>
  <scope>test</scope>
</dependency>
```

- a. Sometimes these dependencies can be test scoped.
2. This will probably break the build at runtime. So remove all the dependencies on `org.sonatype.aether`.
3. The build might or might not be broken at this point, depending on whether the old sonatype dependencies get pulled in from transitive dependencies. Regardless you now need to remove them from the java source code.
4. Search for `import org.sonatype.aether` in your IDE.
5. Some of these classes can be replaced with `org.eclipse.aether` and you're good. However `org.sonatype.aether.impl.internal.SimpleLocalRepositoryManager` is a problem since it's no longer public Maven Resolver.



## Related articles

- [Migrating from org.sonatype.aether to Apache Maven Resolver](#)