

Maven and Eclipse

Importing Maven projects in Eclipse

See [Maven Eclipse Plugin](#)

Editing Maven projects in Eclipse

See [Maven Eclipse Intregation](#)

Building Eclipse plugins in Maven

Instructions targeting Eclipse 3.2, which uses OSGi bundles as plugin format.

The important files to have into account:

- META-INF/MANIFEST.MF the OSGi manifest
- plugin.properties plugin config
- plugin.xml plugin config
- build.properties binaries and sources definition to build with PDE
- ****/*.jar** dependent libraries, need to be in the source folder for building with PDE. They are packaged in the same folder in the zip. eg. if jars are in x/y they end in x/y folder of the jar, PDE doesn't allow changing it.

Currently there are two integration points that overlap:

- [Maven Eclipse Plugin](#) generate manifest and enable PDE nature
 - Need a install plugins in eclipse workbench goal, installing the plugins as folders or jars
- [Apache Felix OSGi plugin](#) can create an OSGi bundle, packaging dependencies, generating manifest,...
 - It always packages the dependencies, it'd need to allow setting Require-Bundle and not package some of them. If using provided scoped dependencies the transitive dependency part doesn't work. Right now it uses Import-Packages instead of Require-Bundle.
 - would need to generate manifest and copy dependencies in working folder so PDE graphic tools can use the,

An OSGi bundle can be easily created with the Felix OSGi plugin. Integration between Maven and PDE is other story.

- manifest needs to be in `${basedir}/META-INF/MANIFEST.MF`
- jars need to be somewhere in `${basedir}`
- other plugin dependencies must be either opened in eclipse or installed in the workspace

This is a configuration that works

```
<build>
  <resources>
    <resource>
      <directory>${basedir}</directory>
```

```
        <includes>
            <include>plugin.xml</include>
</include>plugin.properties</include>
        </includes>
        <targetPath>/</targetPath>
    </resource>
</resources>
<plugins>
    <plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-eclipse-plugin</artifactId>
>
        <configuration>
            <pde>>true</pde>
        </configuration>
    </plugin>
    <plugin>

<groupId>org.apache.felix.plugins</groupId>

<artifactId>maven-osgi-plugin</artifactId>
    <version>0.8.0-SNAPSHOT</version>
    <executions>
        <execution>
            <goals>
                <goal>osgi-bundle</goal>
            </goals>
        </execution>
```

```
        </executions>
    </plugin>
</plugins>
</build>
<pluginRepositories>
    <pluginRepository>
        <id>apache.org</id>
        <name>Apache Snapshot
Repository</name>

<url>http://people.apache.org/repo/m2-snapsh
ot-repository</url>
        <releases>
            <enabled>>false</enabled>
```

```
</releases>  
</pluginRepository>  
</pluginRepositories>
```

Reading Eclipse Plugins / OSGi bundles for Maven

Latest version of [Maven Eclipse Plugin](#) has a [MakeArtifactsMojo](#) that reads Eclipse Plugins, parses the manifest to generate the poms and installs them in the repository.

Building Eclipse with Maven

Several barriers have been identified that prevent Eclipse platform, equinox, RCP from building using Maven. For a discussion, see [Barriers to Building Eclipse with Maven](#).

General Discussion on Building OSGi

I'm starting an area on the MAVENUSER confluence space to talk about how to build OSGi bundles (with Maven). I'd like to capture the current practices, whether using Ant or whatever, and use the space for reviewing possible design changes for Maven to accommodate these builds. That area will be found here:

[Building OSGi Bundles](#)