

Application Composer Maven Plugin

This plugin has two goal:

- `applicationcomposer:run`: to start the application from mvn command line
- `applicationcomposer:zip`: to package a zip with dependencies and start scripts

IMPORTANT

the dependencies are retrieved with `MavenProject.getArtifacts()` which means you artifacts should be a `war` - maven doesn't populate it with a `jar` - and the compile phase - at least - should be passed to ensure it is populated.

Run goal configuration

```
mvn process-classes applicationcomposer:run -DskipTests
```

| Name | Default | Description |
|-------------|---|--|
| args | - | a list of application arguments |
| application | - | application qualified name |
| binaries | <code>\${project.build.outputDirectory}</code> } | where is your module code (target/classes) |
| mavenLog | true | force to use maven logging in openejb |

Zip goal configuration

```
mvn process-classes applicationcomposer:zip -DskipTests
```

| Name | Default | Description |
|-------------|--|--|
| workDir | <code>\${project.build.directory}/\${project.build.finalName}-applicationcomposer</code> | where the container can "work" and create temp files |
| zip | <code>\${project.build.directory}/\${project.build.finalName}-applicationcomposer.zip</code> | where to create the zip |
| attach | true | attach the created artifact |
| classifier | - | artifact classifier if needed |
| application | - | application qualified name |

| Name | Default | Description |
|----------|---|---|
| binaries | <code>\${project.build.outputDirectory}</code> } | where is your module code (target/classes) |