Maven and AEM – Building an AEM Project

Tyler Maynard
AEM DEVELOPER

@TylersDesk www.tylermaynard.com

Overview

- Package development lifecycle
- Maven
- Pom.xml
- Profiles
- Artifacts – Jars, poms and content-packages
- AEM specific archetypes

Package Development Workflow

Collaborating in AEM Projects

Kass – Java Developer

Project Source Code

Ryan – UI Developer

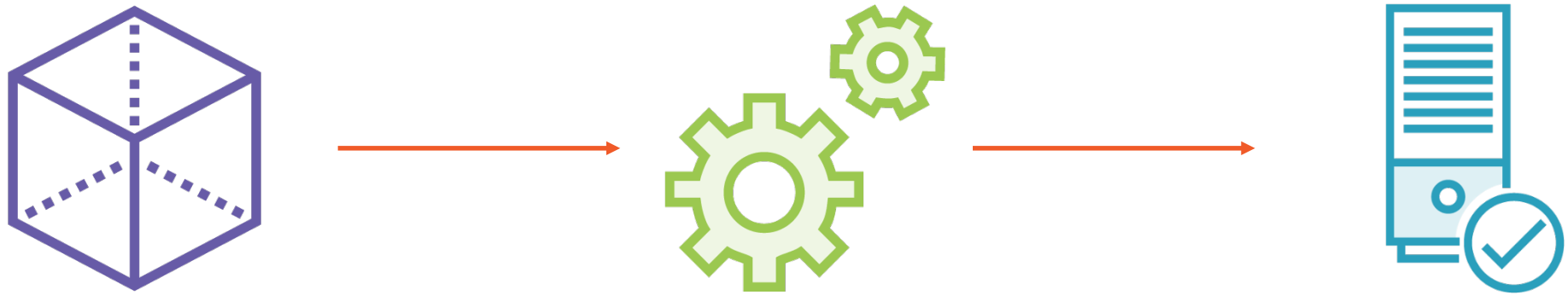
Package Development Workflow

Package project folder

Maven

AEM Package

Package Delivery During Development

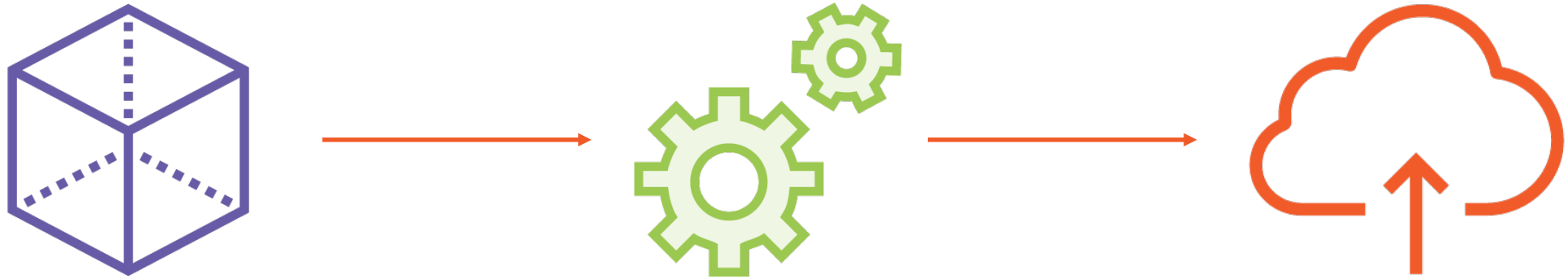


AEM Package

Maven

AEM Instance

Package to Repository

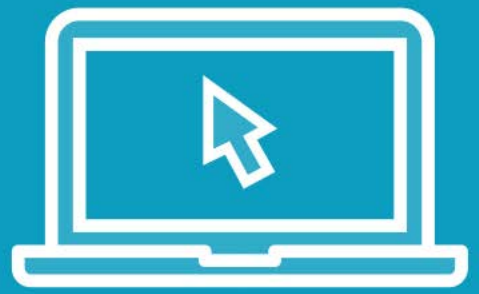


AEM Package

Maven

Repository

Demo



Navigate to a project

Build the artifact

Rebuild and deliver the artifact

Maven - A Build Tool

Maven

A tool that can be used for build automation and managing any Java-based project.

Maven and AEM

Simplifies and standardize builds

Provides project structure

Builds our jars, bundles and AEM packages

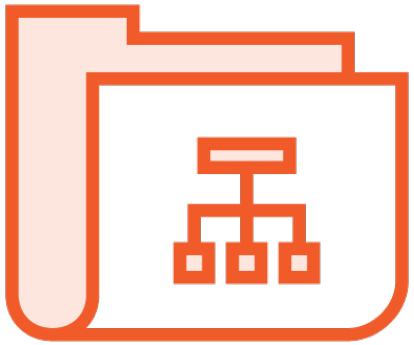
Deploy build artifacts to AEM instances

Maven Concepts

Convention

Configuration

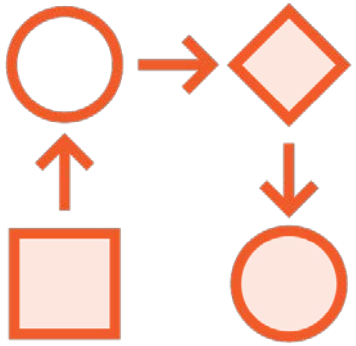
Maven Concepts Continued



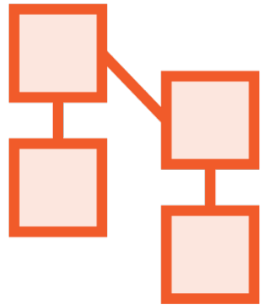
Project Object Model
(pom.xml)

Plugins
(Defined in pom.xml)

Maven Concepts Continued

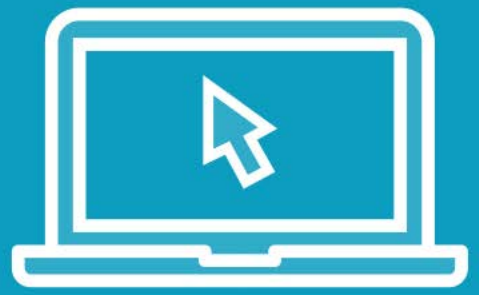


Build lifecycle



Dependency management

Demo



View a Maven project structure

Open the parent pom.xml

Show the artifacts after a build

Pom.xml

Project Object Model

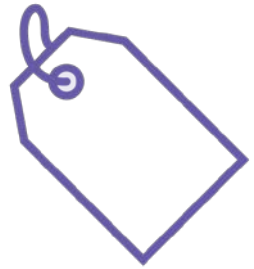
The declaration of a project that includes all necessary information and configuration.

POM Requirements

<modelVersion>

<groupId>

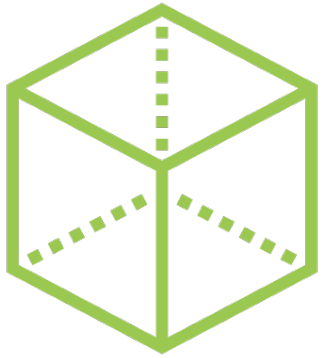
<artifactId>



<version>

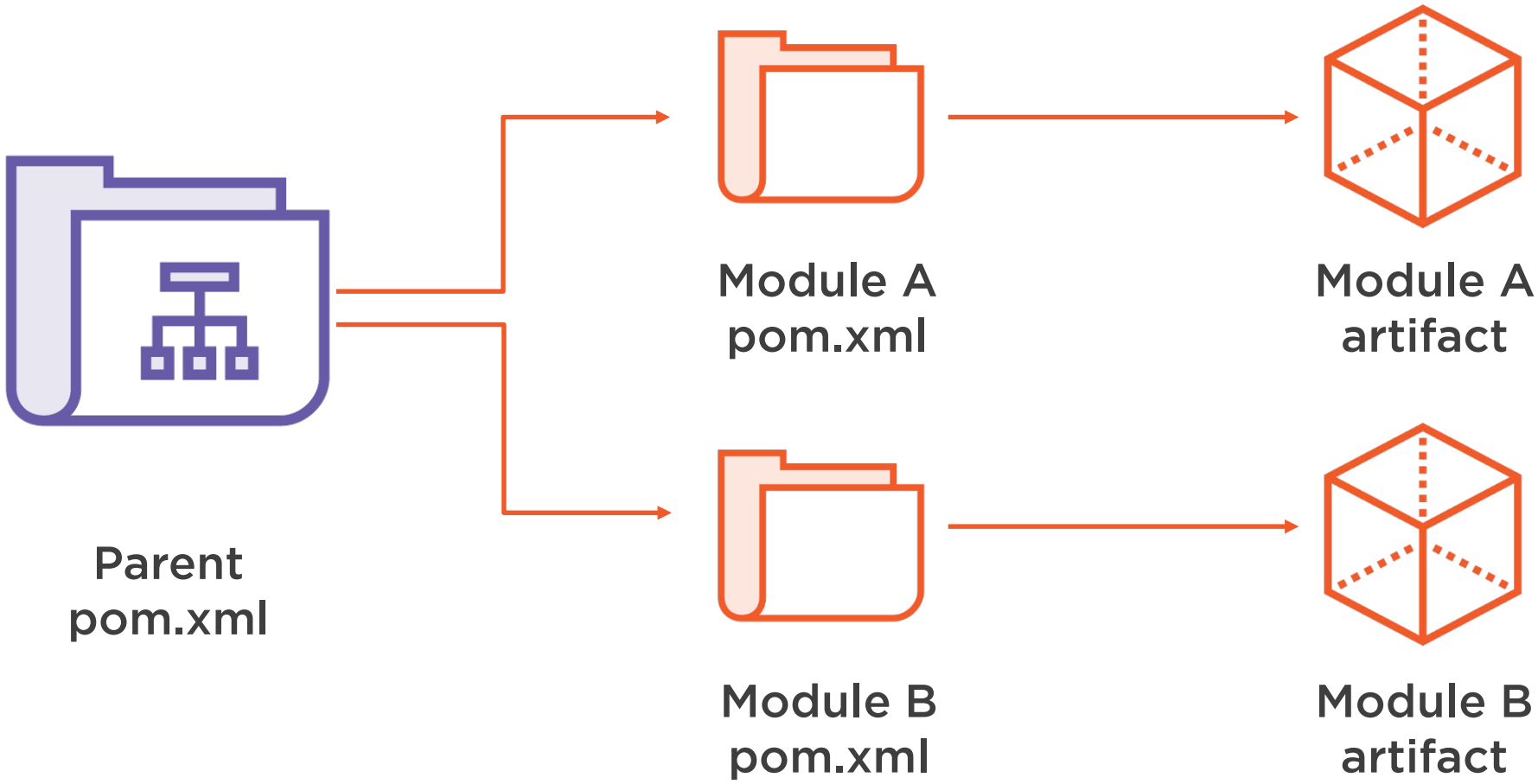
POM Packaging

<packaging>
pom
</packaging>

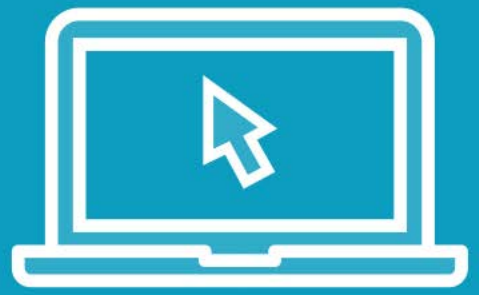


<packaging>
content-package
</packaging>

Parent POM


Parent
pom.xml
Module A
pom.xml
Module A
artifact
Module B
pom.xml
Module B
artifact

Demo



- Open parent pom.xml
- Investigate maven coordinates
- Packaging
- Investigate sub module

Maven – Properties, Plugins and Lifecycles

Custom Maven Properties

```
<properties>
    <property.name>value</property.name>
    <another.property>value</another.property>
</properties>
```

Accessing Maven Properties

```
<element>
    ${property.name}
</element>
```

General Maven Properties

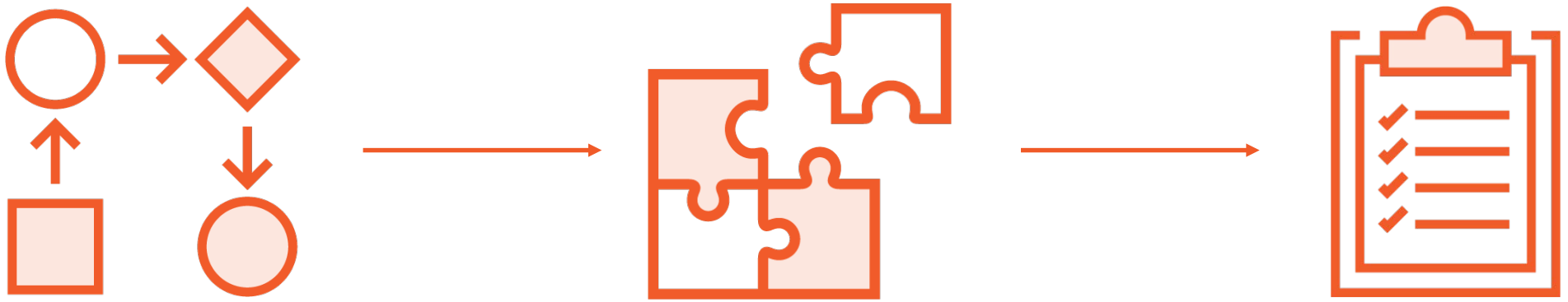
```
${project.*}
${basedir}
${env.*}
${settings.*}
```

Maven Plugins

Plugin

Goals

Maven Lifecycles Overview

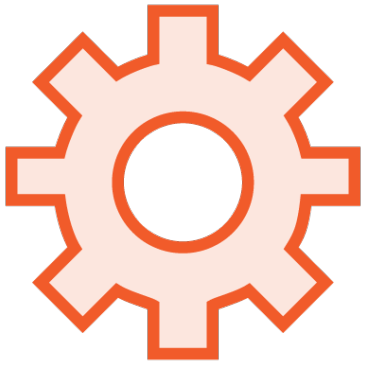


Maven Lifecycle

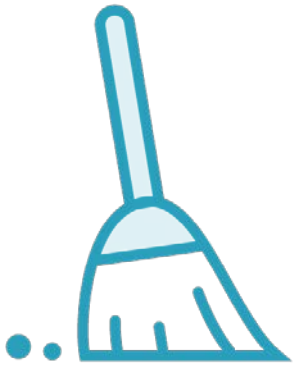
Phases

Plugin goals at specific phases

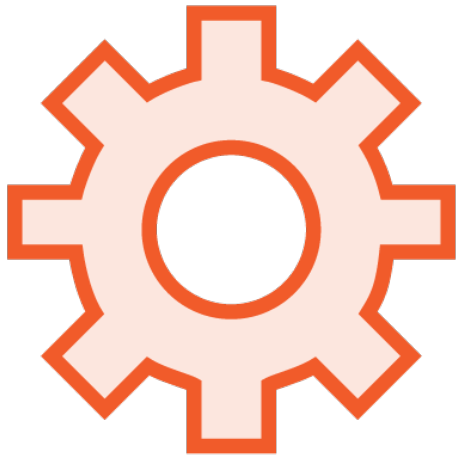
Lifecycles for AEM Development



Default (build)



Clean



Validate

Compile

Test

Package

Verify

Install

Deploy

Build Phases for AEM Development

mvn package

mvn install

Using a Plugin

```
<plugin>
    <groupId></groupId>
    <artifactId></artifactId>
    <version></version>
    ...
</plugin>
```

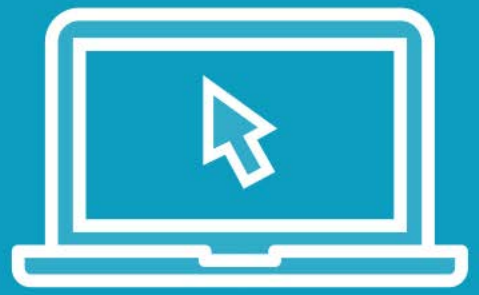
Executing a Plugin Goal

```
<plugin>
    ...
    <executions>
        <execution>
            <goals>nameOfGoal<goals>
        </execution>
    </executions>
    ...
</plugin>
```

Configuring a Plugin

```
<plugin>
    ...
    <configuration>
        <plugin-property>value</plugin-property>
        <plugin-property>value</plugin-property>
    </configuration>
</plugin>
```

Demo



Custom properties

Plugins

Run clean lifecycle

Run build lifecycle

Profiles, Repositories and Dependencies

Build Profile

A defined configuration that can be used to override build defaults or even extend a build.

Profile Nodes

```
<profiles>
    <profile>
        <id>name</id>
        <activation>...</activation>
        <build>...</build>
    </profile>
</profiles>
```

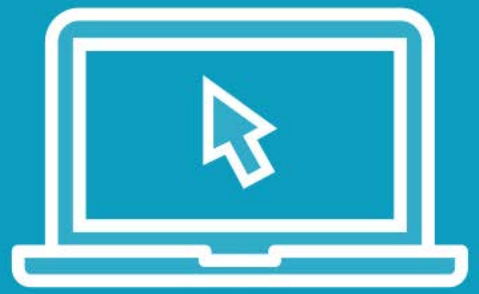
Activating Profiles

```
-P[id of profile goes here]
```

Example:

```
mvn install -PautoInstallPackage
```

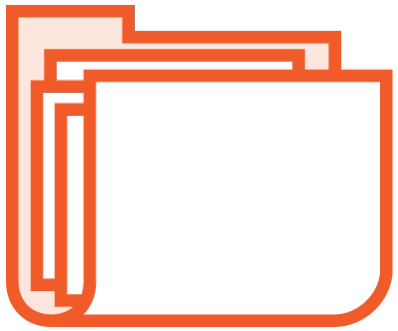
Demo



View the autoInstallPackage profile

Run the autoInstallPackage profile

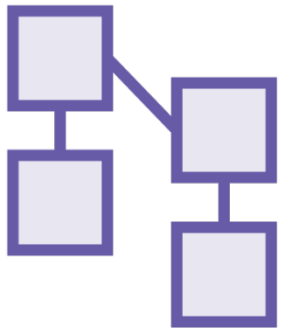
Maven Repositories



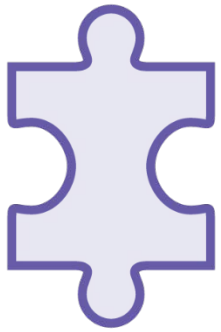
Local Repository

Online Repository

Artifacts in Repositories



Dependency Artifacts



Plugin Artifacts

Dependency Management

```
<dependency>
    <groupId>javax.jcr</groupId>
    <artifactId>jcr</artifactId>
    <version>2.0</version>
    <scope>provided</scope>
</dependency>
```

Local Repository Location

Mac/Linux:

/Users/<yourUser>/.m2

Windows:

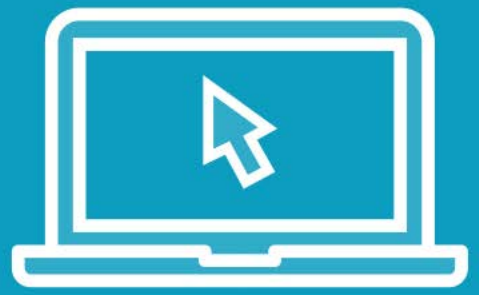
C:/Users/<yourUser>/.m2

Remote Repositories

```
<repository></repository>

<pluginRepository></pluginRepository>
```

Demo



View dependencies in the pom.xml

Navigate to local repository

Locate simple-project artifact in repository

Maven Archetypes

Maven Archetype

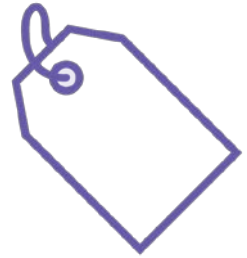
Maven project blueprint or template.

Archetype Properties

Archetype repository

GroupId

ArtifactId



Version

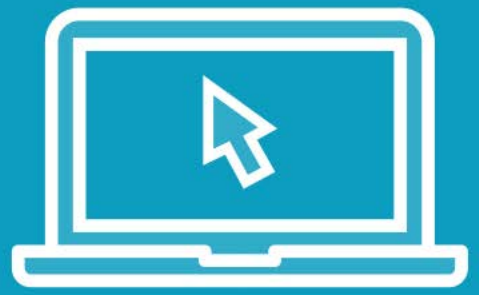
Generating Project From an Archetype

```
mvn archetype:generate -DarchtypeRepository=<repo>
    -DarchetypeGroupId=<groupId>
    -DarchetypeArtifactId=<artifactId>
    -DarchetypeVersion=<version>
```

AEM Archetype Catalog

https://repo.adobe.com/nexus/content/groups/public/archetype-catalog.xml

Demo



View the archetype catalog

Select an archetype

Generate project from archetype

Investigate project structure

Build Archetype

Summary

- Package development workflow
- Maven
- Pom.xml
- Artifacts
- Profiles
- Repositories
- Archetypes