Multi-Modules

Thursday, June 7, 2018　　9:40 PM

Large Software Application

- Multiple modules
  - Multiple teams
    - different modules
    - Same application

Demo2 Team

- Demo.jar
  - Demo team builds new jar
    - Inform every time when the Code is updated
  - Demo2 have to update POM.xml to get the latest Demo.jar

Snapshot

- Special Version
  - indicate current development
- Maven checks for new snapshot version in a remote repository for every build

→ 1st class Multi-module support

→ Each Maven project creates 1 primary artifact.

→ A parent POM.xml is used to group projects
  - Call the parent
    - It calls the child

$$P \underset{}{\overset{}{\Big\langle}} \begin{array}{l} C \\ C \\ C \end{array}$$

→ To define a Parent POM.xml
  - Packaging is changed to <u>POM</u>

  ... are given

├ packaging is changed to **pom**

├ References to Child are given

<module>
    <module> Child 1 </module>
    <module> Child 2 </module>
</modules>

→ In case of modules having dependencies
and to form a build order

├ The child's should include a parent tag

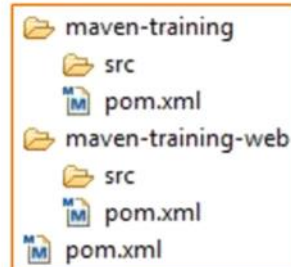├ This way the parent reads the child POM and creates a build order

→ To build only one module

├ mvn clean -pl <module name>

$ mvn clean –pl <modulename>

```
<project>
  <groupId>EBU</groupId>
  <artifactId>Parent-module</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>pom</packaging>
    <modules>
        <module>Child-jar</module>
        <module>child-war</module>
    </modules>
</project>
```
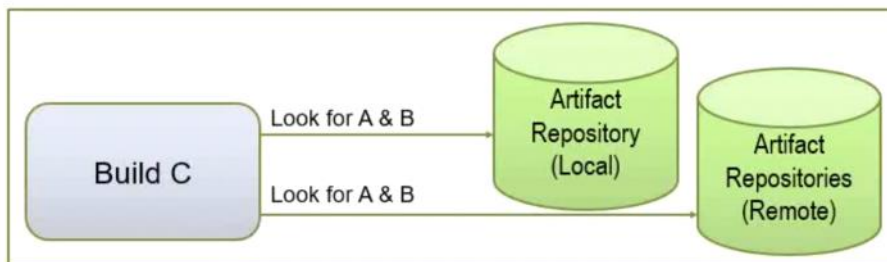
📂 maven-training
  📂 src
  Ⓜ pom.xml
📂 maven-training-web
  📂 src
  Ⓜ pom.xml
Ⓜ pom.xml

Multi Modules ..

```xml
<project>
...
<parent>
    <groupId>EBU</groupId>
    <artifactId>Parent-module</artifactId>
    <version>1.0-SNAPSHOT</version>
</parent>
<groupId>EBU</groupId>
<artifactId>child-jar</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>jar</packaging>
...
</project>
--------------------------------------------------------------------------------
<project>
...
<parent>
    <groupId>EBU</groupId>
    <artifactId>Parent-module</artifactId>
    <version>1.0-SNAPSHOT</version>
</parent>
<groupId>EBU</groupId>
<artifactId>child-war</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>jar</packaging>
...
</project>
```
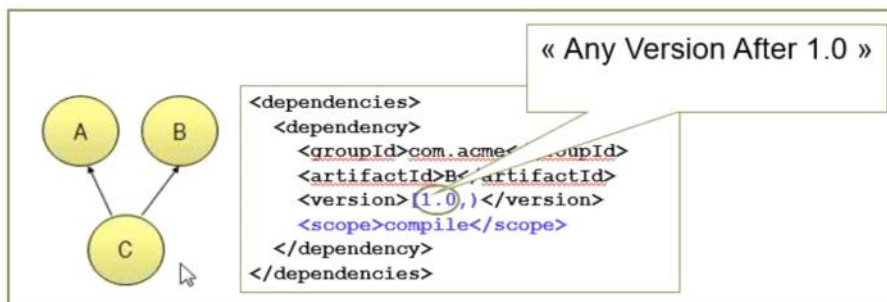
Dependency Management

« Any Version After 1.0 »

```xml
<dependencies>
    <dependency>
        <groupId>com.acme...oupId>
        <artifactId>B</artifactId>
        <version>[1.0,)</version>
        <scope>compile</scope>
    </dependency>
</dependencies>
```

A B

C

Build C → Look for A & B → Artifact Repository (Local)

Look for A & B

Artifact Repositories (Remote)

Transitive Dependencies

- Transitive Dependency Definition:
  - A dependency that should be included when declaring project itself is a dependency
- ProjectA depends on ProjectB
- If ProjectC depends on ProjectA then ProjectB is automatically included
- Only compile and runtime scopes are transitive