Maven Build Profiles

- Maven Build Profiles allow you to specify a set of build configuration values
- Values you can specify are largely the same data elements found in the POM
- Values set in build profiles:
  - Can be in addition to what is in the project POM
  - OR used to override POM values
- More than one profile can be set to active at time
- **Caution**: No priority is available for multiple active profiles.
  - Duplicate Property resolution is random

Why Use Build Profiles?

- Build Profiles are a very powerful feature which allows you to shape the behavior of your build
- Build Behavior can be changed:
  - Based on your desired outcome of the build
  - Automatically based on the run time environment
  - Automatically to adopt to different operating systems
  - To optionally activate specific plugins
  - To provide alternative build time configuration values

Declaring Build Profiles

- **Per Project**
  - Defined in pom.xml
  - Command Line - mvn package -S <path to settings file>
- **Per User**
  - Defined in <user home>/.m2/settings.xml
- **Global**
  - Defined in <Maven Home>/conf/settings.xml

Declaring Build Profiles

- **Profile Descriptor**
  - Defined in profiles.xml
  - Not Supported by Maven 3.0 (Released in 2010)
  - profiles.xml will not be covered in this course
  - Documentation is available in the archive section on the Apache Maven website

Which Declaration Method to Use?

- Use Project POM when:
  - Your build needs to be portable to other computers (ie CI server, other developers)
  - When others may or may not need build features
- Use Settings.xml when:
  - You want the profile available to many projects
  - Frequently used for configuration of repository managers
  - Protection of secrets
  - Environment Config values

Combo of Declaration Methods?

- Can you define profile values in both POM and settings.xml for the same profile?
  - Short answer is yes. . .
  - Not recommended
  - Considering using multiple profile id’s

Activating Build Profiles

- In the Profile configuration under activation attribute:
  - setting activeByDefault property to true
  - default activation for OS, JDK versions
  - existence of system properties
  - specific values of system properties
  - missing files (ie build artifact not in target dir)
- Command line: mvn package -P <profile-1>,<profile-2>
- In settings.xml file - activeProfiles section

Deactivating Profiles

- Profiles can be deactivated from the command line:
    - mvn package -P !<profile-1>,!<profile-2>
    - mvn package -P -<profile-1>,-<profile-2>

POM Elements in Profiles

- repositories
- pluginRespositories
- dependencies
- plugins
- properties
- modules
- reporting
- dependencyManagement
- distributionManagement

POM Elements in Profiles

- Build Element: (Only)
  - defaultGoal
  - resources
  - testResources
  - finalName

View Active Profiles

- Command: mvn help:active-profiles
- Note should be used with relevant CLI options
  - mvn help:active-profiles -P <profile-1>
    - would show profile-1 active