Taking advantage of Apache Maven for Android development

Manfred Moser

simpligility technologies inc.
http://www.simpligility.com
@simpligility

simpligility technologies inc

Audience Background

Android development

Eclipse

Apache Ant

Apache Maven

simpligility technologies inc

About Manfred

- Long time Linux user and Java developer
- Currently working as Android application developer
- Co-Author of the book Maven: The Complete Reference
- Committer on Maven Android Plugin
- Author of Maven Android SDK Deployer
- Vancouver Island Java User Group Founder/Leader

Agenda

Motivation

Brief Apache Maven introduction

Maven Android tool chain with examples

Lots of tips and tricks

Q & A

Building Android Apps Currently

Eclipse ADT

Or

Apache Ant based build

Whats wrong with that?

You don't like Eclipse or Ant or don't want to depend on it for build.

You need additional features in the build.

You want command line and continuous integration server usage.

You need to work with multiple dependencies.

You want to reuse code from another project.

Example Dependencies

Make “libs” folder and copy jars files into it.

Check them into version control.

Upgrade means replace jar file and transitive dependencies.

Example Dependencies

Known to be a hassle from years of Apache Ant usage on Java projects:

- Unknown dependencies
- Transitive dependencies
- No documentation
- No collision detection
- Large project checkout

Led to Apache Maven, Apache Ivy ...

Introduction to Apache Maven

"Software project management
and comprehension tool"

Builds your software and much more

De-facto standard for Java software builds

Convention over configuration (like RoR)

Declarative rather than procedural

Maven and Android

Maven Android Plugin – core tool

M2Eclipse, M2Eclipse Android Integration – for Eclipse users

Maven RIndirect Plugin – for reuse/library setup

Maven Android SDK Deployer, Android4Maven, Android SDK Tool – all optional for setup

Maven Installation

Unzip, create M2_HOME, add to PATH

apt-get install maven2

port install maven3

and then just run

mvn -version

Getting started

Convert your existing app – add pom.xml file

Create new project – use command line or IDE wizard to create normal Android project, then add pom.xml

Use Maven Archetype – on command line or in IDE wizard

Hello Flash Example – pom.xml

Packaging apk

Android dependency

Java source folder

Maven Android Plugin configuration

Maven Compiler Plugin

Doing a build

mvn clean install

Invokes the **clean** and the **install**

build life-cycle phase

Phases

- predefined order of tasks for build
- what happens is defined in pom (+ super-pom)
- additional plugin goals can added

pre-clean, clean, post-clean
validate, generate-sources, process-sources, generate-resources, process-resources, compile, process-classes, generate-test- sources, process-test-sources, generate-test-resources, process- test-resources, test-compile, test, prepare-package, package, pre-integration-test, integration-test, post-integration-test, verify, install, deploy

Starting the Android emulator

mvn android:emulator-start

Invokes the
emulator-start **goal**
of the
maven **android** plugin

Plugins

Maven itself does nearly nothing
– just a container (M2 - Plexus, M3 - Guice)

Super POM defines standard configuration
which can do a LOT

Lots of plugins available at apache, codehaus and beyond

Goals

different per plugin
behaviour can be defined in pom
e.g. mvn install:install

but does not have to be

mvn archetype:create ... mvn install:file ....
pass parameters in with -Dparameter=value

Maven Invocation

mvn [options] [<goal(s)>] [<phase(s)>]

- Options – get list with mvn -h

- Goals – with syntax Plugin:PluginGoal
- e.g. mvn android:deploy

- Phases – e.g. clean compile test install package
- e.g. mvn clean install

Downloading the internet? Not!

Core install of Apache Maven very small (3mb)

What is needed gets downloaded into repository..

Repository

Storage for plugins and artifacts

Local user repository ~/.m2/repository

Maven central

Other public repositories

Proxy repository server

Recap

pom.xml

Default build life-cycle with phases

Plugins with goals

Repositories

Next - Use Cases

For Android Development

Using external dependencies

Add dependency

Everything else happens “automagically”

Roboguice – Astroboy example

Other external dependencies

Google Guice IoC
KSOAP2-Android for SOAP parsing
Jackson JSON Parser
Robolectric
Robotium
GoogleAPI's
and so on and so on

How did it find android.jar?

Deployed to Maven Central repository

Built from AOSP with Android4Maven

Google Maps jar

Can't be in Maven central

Use Maven Android SDK Deployer

Copies jars from SDK install to repository

Maps Dependency

```
<dependency>
  <groupId>com.google.android.maps</groupId>
  <artifactId>maps</artifactId>
  <version>7_r1</version>
  <scope>provided</scope>
</dependency>
```

Dependency Scopes

Default = compile

provided

test

Scala

special use of a library jar

enables Scala programming language

scala plugin for compilation

proguard for shrinking

Scala example

Automate version number

pom.xml authoritative version

managed by Maven

resource filtering populates Android manifest

Morseflash example

Automate version code

Auto-generated, increasing integer

Based on time-stamp

Morseflash example

Maven resource filtering

Define properties or use an existing one

Define filtered resources

Embed properties for replacement

Can also be used to exclude files (e.g. SVG source for images)

Development vs Production

Resource filtering with different values for properties

Depending on build time switch (e.g. automatic or provided at invocation)

Uses profiles

Morseflash example

Profiles

Separate build definition
e.g. properties
but also
plugin configuration/execution/...

Activation via switches, files, OS...

Proguard path configuration

rt.jar/classes.jar and jsse.jar

different path/filename for different OS

Morseflash example

Sign apk

Maven Jarsigner Plugin

keys, password ... settings.xml

Morseflash example

Zipalign

Maven Android Plugin

Morseflash example

Proguard

Obfuscate, shrink and optimize

Only in release profile

Morseflash example

Releasing Application

Perform release with Maven Release Plugin

Will handle scm operations, version changes...

Take advantage of repo server for artifact storage

MorseFlash Example

Pull and Push

android:pull and android:push goals

Upload database or image resources to device or emulator avd

Download performance .trace files

Deploy/Undeploy

Remove old version of application before testing

Install application

Goals of Maven Android Plugin

Deploy any apk

- Deploy an application as part of e.g. preparing avd for ci build
  - Droidreader for PDF testing
  - OpenOffice document reader for odf
- Example droidreader apk install (android.file parameter)

Screen size, platform level

emulator-start, emulator-stop

different profiles

different parameters for deploy

deploy to different emulators/devices

License headers

Maven License Plugin

all source code - license and copyright header

Produce project website

Maven site plugin with javadoc and so on

Sonar for historical trending and more

RoboGuice example

Static code analysis

As part of build

Findbugs
PMD
Checkstyle

Reuse projects

Plain java library project from server side application

Android projects – dependency type “apksource”

Android projects – Maven RIndirect Plugin

Android library projects – support coming soon

Android NDK .so files – as normal dependency

Continuous Build

Run on any operating system

Including VMs in the cloud

Run full build

More details next presentation

Testing

Unit & Integration Testing

Various tools

All as part of build

More details next presentation

Related Tools

- Android4Maven – builds jars in Maven central
- Maven Android SDK Deployer – maps jars into repo
- Android SDK Tool – install SDK headless

Your Android build needs?

Easy to script more features (Ant, Groovy,...)

All open source tools

Maven Resources

Apache Maven
http://maven.apache.org

Maven: The Complete Reference
and other books and resources
http://www.sonatype.com/book

M2 Eclipse
http://m2eclipse.sonatype.org/

Maven Android Resources

Maven Android Developers Mailing List
http://groups.google.com/group/maven-android-developers

Maven Android Plugin
http://code.google.com/p/maven-android-plugin/

Maven RIndirect Plugin
http://www.github.com/akquinet/android-rindirect

M2 Eclipse Android Integration
http://code.google.com/p/m2eclipse-android-integration/

Examples

Samples from Maven Android Plugin
http://github.com/mosabua/maven-android-plugin-samples

RoboGuice Sample
http://code.google.com/r/mosabua-roboguice/source/browse

Summary

Proven, ready alternative

More flexibility and power out of the box

Better for continuous integration

Better reuse of components and in team environment

The End

Thank you for your attention.

@simpligility

manfred@simpligility.com