

JavaFX on Android



Johan Vos

johan@lodgon.com

@johanvos

JavaOne 2014, [CON 1804]



Why JavaFX?



- Write your JavaFX Application once, deploy it on as many platforms as possible, reaching a larger audience.
- Cost-efficient: one platform for creating apps on desktop, iOS and Android.
- By design, JavaFX is a powerful, modern framework, with input from a number of other UI frameworks.
- JavaFX is Open Source. The development is done in the open.

Why on Android?



- Mobile devices are now more popular than desktop/laptops
- On mobile devices, apps usage outnumbers mobile browser usage
- Mobile apps are changing the IT industry
- Apps on mobile devices = {iOS, Android, ...}
- Clients provide the access point to back-end servers, where Java is leading the pace. A business model on the back-end benefits from a client strategy

History



- OpenJFX contains code for iOS and Android
- No official Oracle release
- Community stepped in and helped
- JavaFX 8u20 almost same source code on desktop and Android
- We just started
- Thanks to Tomas Brandalik and Stefan Fuchs and the OpenJFX Team

Demo



Internals



- Runs on top of Dalvik
- JavaFX native code is cross-compiled to ARM
- JavaFX Java code runs on Dalvik
- A compat.jar library is added since the JavaFX Java code requires more than what Dalvik provides
- Launcher and interaction classes are in jfxdvm.jar
- Build tools are provided to bundle your application and the Dalvik SDK together

Internals



- Download dalvik-sdk-8u20 from <https://bitbucket.org/javafxports/android/downloads>
- Structure dalvik-sdk directory:
 - rt/lib/ext: jar files
 - rt/lib/armeabi: native libraries
 - android-tools and samples: build scripts and tools

Build tools



- gradle-ant based
- complicated
- working on it
- samples directory shows how to do it
- 3 steps:
 - create your JavaFX Application
 - use a gradle script to create an Android project based on your JavaFX Application
 - compile the created Android project into an Android package

Your JavaFX Application



- A regular JavaFX Application
- use Java 7 APIs
- no support for Java 8 APIs except for Lambdas
- make sure a jar file is created, and the required dependencies are in the jar file or in the same directory as the jar file
- gradle handles this very well

Demo



HelloWorld

Create Android Project



- We need
 - location to Android SDK (download)
 - location to JavaFX-Android SDK (download)
 - location to our JavaFX application
- Gradle script will create an Android project based on our values

Demo



Step 3: Create Android APK



- the gradle-script created a directory with an ant build file, and it refers to additional files in the JavaFX-Android SDK.
- a debug/release version of the package will be created using
 - `ant clean debug`
 - `ant clean release`
 - transfer this to your device using
`adb install`

Demo



WheresDuke



- OpenMapFX: open-source project with JavaFX based map components.
 - <http://bitbucket.org/lodgon/openmapfx>
 - see Java Magazine
- Desktop version
- Leapmotion controlled version
- Android version
 - now branded WheresDuke
 - beta-version in PlayStore
 - if you want access, let me know at johan@lodgon.com
 - layout work is ongoing

Demo



WheresDuke



- More than HelloWorld
- Uses many JavaFX 8 features
 - WebView
 - Controls, Containers
 - CSS styling
- DataFX for communication with GlassFish back-end
- DataFX supports OAuth



- In a JavaFX Android Application, you can leverage Android AND JavaFX services/APIs
- Be careful and consistent
- No Location API in Java SE 8 or JavaFX 8
- LocationService and other services are accessible via the Android Context

```
Context ctx = FXActivity.getInstance();  
Object getSystemService =  
ctx.getSystemService(FXActivity.LOCATION_SERVICE);
```

Integrating Android services in projects



- Write Once, Run Anywhere
- ServiceLoader approach:
 - define an interface
 - desktop and android packages each have their own implementation of the interface
 - META-INF/services directory contains the link to the implementation class
 - bundle either the desktop or the android package at runtime

The Project



- bitbucket repositories: bitbucket.org/javafxports
- website: javafxports.org
- mail discussions: <https://groups.google.com/forum/#!forum/javafxandroid>
- open-source, community maintained initiative that is very well aligned with OpenJFX
- commercial support available, e.g. http://www.lodgon.com/dali/page/JavaFX_Mobile_Consulting
- johan@lodgon.com
- [@johanvos](#)