King Training Resources

Build a Mobile App in 60 Minutes with MAF

Presented by: John Jay King

Download this paper from: OAUG web site
Or http://www.kingtraining.com

Session Objectives

- Understand the components of Oracle MAF
- Use Oracle MAF to create mobile applications for iOS and/or Android
- Learn how MAF provides resources to make application creation simpler

Who Am I?

- John King – Partner, King Training Resources
- Oracle Ace Director
- Member Oak Table Network
- I create, customize, and provide training in Oracle and other topics (http://www.kingtraining.com); and aid customers with new technologies & practices
- "Techie" who knows Oracle, ADF, MAF SQL, Java, and PL/SQL pretty well (along with other topics)
- Member of AZORA, ODTUG, IOUG, and RMOUG
- One of those "dog-spoiling" people

King Training Resources

- Providing customized training solutions since 1988 in the US and internationally
- Oracle topics include: SQL, PL/SQL, Database, Cloud, APEX, ADF, MAF, Forms, Reports, Pro*C/ Pro*COBOL
- Non-Oracle topics include: UX, Web Services, IoT, REST, Cloud Foundry, Java, JavaScript, HTML5, CSS, jQuery, COBOL, .NET, SQL Server, DB2, Business Analyst, more
- Visit us at www.kingtraining.com for more information and free downloads of presentations and code
- Contact Peggy at 1.303.798.5727 to schedule training today (email: peggy@kingtraining.com)

Mention code OAUG17 for 10% discount before Jan 2018

Arizona, USA

Who Are You?

- Application Developer
- Mobile Developer
- DBA
- Business Analyst
- UX Expert
- Solutions Architect

Think Mobile

- Today, most of us reach for our mobile device to accomplish many daily tasks
- Desktop, laptop, tablet, and phone each provide connections with users
  (not to mention personal devices like Apple Watch, Pebble, Samsung Gear, Basis, Fitbit, Jawbone, GoogleGlass, etc.)
- Your users EXPECT to access information and perform normal tasks when mobile
  (if not using your software, then whose?)

Mobile Application Choices

- Native Deploy “native” specific to iOS or Android (or other…)
- Web Deploy as web application; works on pretty much any browser
- Hybrid Vendor provided development environment that deploys to iOS or Android (or other…) but leverages open technologies like JavaScript, HTML5, and CSS

Native vs Web vs Hybrid

- Here's a great comparison done by the Dzone website
  http://java.dzone.com/articles/state-native-vs-web-vs-hybrid

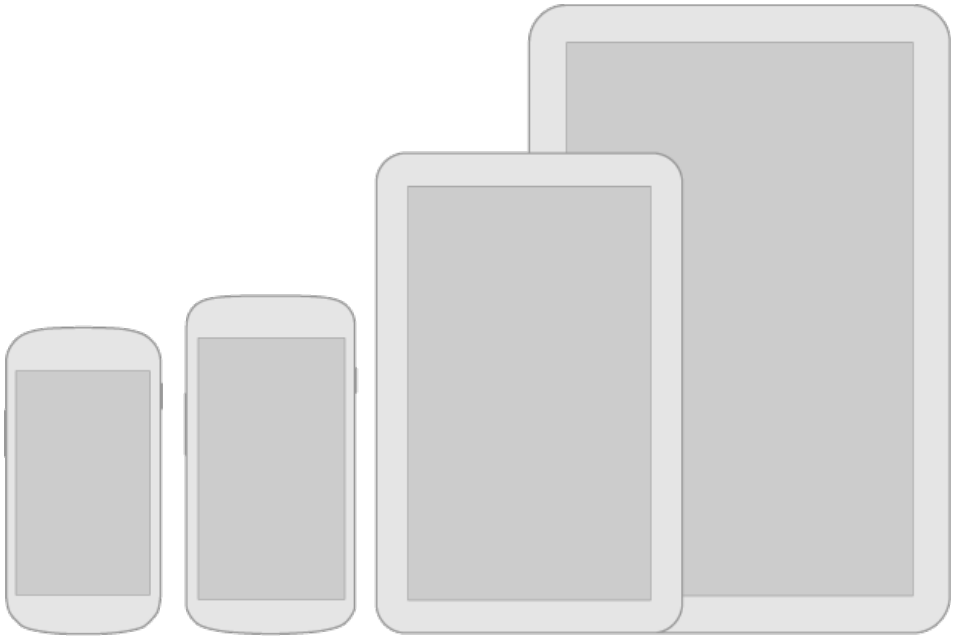
NATIVE vs. WEB vs. HYBRID: 7 FACTORS OF COMPARISON

KEY: CON | PRO | NEUTRAL

	NATIVE	HYBRID	WEB
COST	Commonly the highest of the three choices if developing for multiple platforms	Similar to pure web costs, but extra skills are required for hybrid tools	Lowest cost due to single codebase and common skillset
CODE REUSABILITY/ PORTABILITY	Code for one platform only works for that platform	Most hybrid tools will enable portability of a single codebase to the major mobile platforms	Browser compatibility and performance are the only concerns
DEVICE ACCESS	Platform SDK enables access to all device APIs	Many device APIs closed to web apps can be accessed, depending on the tool	Only a few device APIs like geolocation can be accessed, but the number is growing
UI CONSISTENCY	Platform comes with familiar, original UI components	UI frameworks can achieve a fairly native look	UI frameworks can achieve a fairly native look
DISTRIBUTION	App stores provide marketing benefits, but also have requirements and restrictions	App stores provide marketing benefits, but also have requirements and restrictions	No restrictions to launch, but there are no app store benefits
PERFORMANCE	Native code has direct access to platform functionality, resulting in better performance	For complex apps, the abstraction layers often prevent native-like performance	Performance is based on browser and network connection
MONETIZATION	More monetization opportunities, but stores take a percentage	More monetization opportunities, but stores take a percentage	No store commissions or setup costs, but there are few monetization methods

“Responsive”

- Whether apps are Native iOS/Android, Hybrid, or Web-based; our user’s devices come in all shapes and sizes
- Well-designed apps respond gracefully



Responsive Design

- Think flexibly; layouts should stretch and compress to fit different heights and widths
- Maximize layouts; larger devices should use the extra “real estate” – views should combine multiple sub-views to ease assembly
- Prepare for density differences (DPI); might require resources for different densities

King Training Resources

What is MAF?

- Hybrid approach from Oracle (single code base)
- Free to develop* - JDeveloper or Eclipse (OEPE)
- Runtime Oracle license (per user/per app)
- Free runtime if:
  - You have a current license for Oracle Mobile Cloud Services
  - Your app makes at least one call to Oracle MCS
- No requirement for Oracle ADF or WebLogic licenses
- Replacement for Oracle ADF Mobile

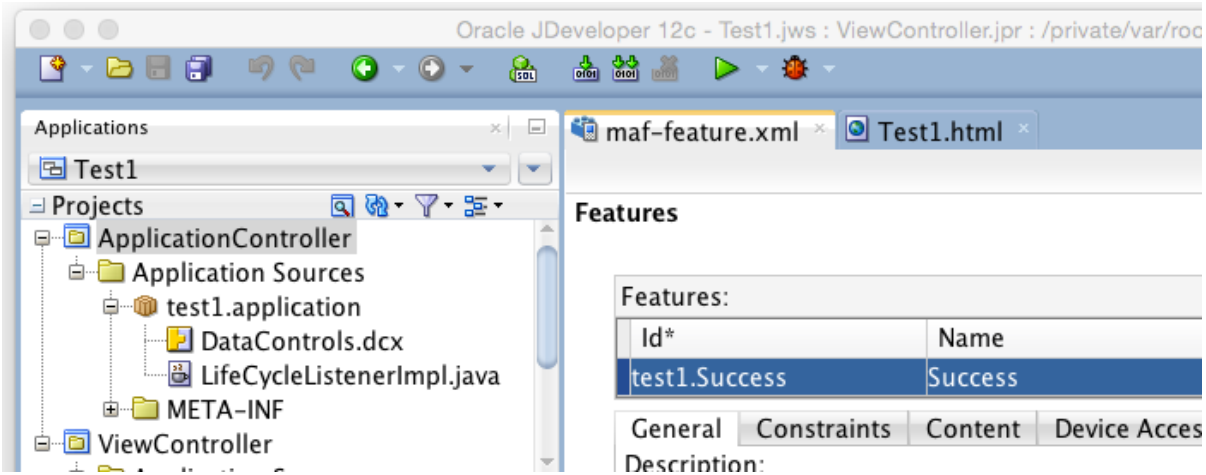
* iOS development requires license from Apple

Why MAF?

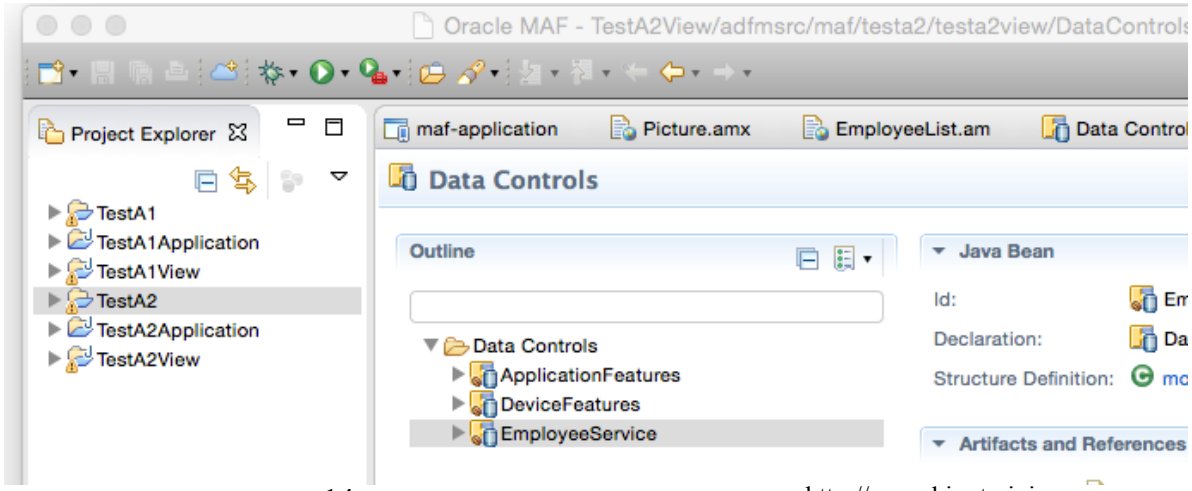
- Single framework for both Android and iOS
- Pre-installed HTML5, JavaScript, and CSS
- Components defined for Apache Cordova (PhoneGap) objects

MAF Development Tools

- JDeveloper



- Eclipse (Oracle Enterprise Pack for Eclipse – OEPE)



Non-Oracle Tools Needed

- Java SDK 
- Android 
  - Android SDK
  - Android Simulator
  - Intel HAXM (Hardware Accel. Exec. Manager)
- iOS 
  - Apple Developer account
  - Xcode
- Oracle MAF Certification Matrix
  http://www.oracle.com/technetwork/developer-tools/maf/documentation/mafcertification-2218073.html

Working with Data in MAF

- Bindings and Data Controls
- MAF Model Layer
- Business Services
  (REST; no SOAP after 2.3)
- Managed Beans
- SQLite (via JDBC)
- ADF BC Objects

MAF User Interface

- Views
- Task Flows
- HTML5 + JavaScript + CSS
  – Touch, Gestures, Responsiveness
- Apache Cordova (PhoneGap)
  – Device Features

MAF App Basics

- Eclipse OEPE
  - App Project
  - App Model
  - App View
- Jdeveloper
  - ApplicationController
  - ViewController

Creating MAF App - Design

1. Watch User at Work; what do they need?
2. Design application (wireframe) to minimize interfaces and maximize impact
3. Determine data required for display, input, and output

Creating MAF App, page 1

1. Create MAF project
2. Add Features to app
3. Create main MAF Task Flow & Empty Views
4. Create Data Controls and supporting Model components
5. Create and test UI
   a) Simple functionality?
   b) Ease of use?
   c) Get job done in few steps?

Creating MAF App, page 2

6. Enable and test mobile features
   a) Visualizations
   b) Camera
   c) GPS
   d) Email
   e) Text
   f) Address Book
   g) Web Services
7. Prepare for Deployment
8. Deploy

Creating MAF App, page 3

9. Deployment Steps
    a) Create Deployment Profile
    b) Deploy to Android Simulator
    c) Deploy to iOS Simulator
    d) Deploy to live Android
    e) Deploy to live iOS
        i. Deploy to file
        ii. Import file into iTunes
        iii. Sync with device
10. Test on all likely devices

Path to Deployment

- Deployment is NOT simply click Android/ iOS and go…
- Steps include:
  - Prepare for deployment by setting up Android and iOS targets in JDeveloper/OEPE
  - If deploying to iOS device; device must be registered and the developer must have created a provisioning profile previously

Deploy - Android Emulator

- Deploy and Test – Android Emulator 
  - Command line; adb start-server (Android debug server found in <androidhome>/platform-tools)
  - Start AVD Manager; select simulator
  - In Eclipse: Debug->Debug Configurations
    - MAF Application “New icon”
    - Provide name; Choose project, choose Target, refresh; DEBUG
  - Open app in emulator

Deploy – iOS Simulator

- Deploy and Test – iOS Simulator 
  - In XCODE: Xcode->Open Developer Tool ->iOS Simulator
  - In Eclipse: Debug->Debug Configurations
    - Choose MAF application name
    - Choose target, refresh, DEBUG
  - Open app in emulator

Deploy – Android Device

- Deploy and Test – Android Device

  - Connect device via USB
  - Command line:
    adb usb (might need to adb kill-server first)
  - In Eclipse: Debug->Debug Configurations
    - Choose MAF application name
    - Choose target, refresh, DEBUG
- Open app in device

Deploy – iOS Device

- Deploy and Test – iOS Device 
  - Connect device via USB
  - Start iTunes (if not started automatically)
  - In Eclipse: Debug->Debug Configurations
    - Choose MAF application name
    - Choose target, refresh, DEBUG
  - If successful build; open iTunes and select device to be deployed to
  - In ITunes: File->Add To Library; select .ipa file
  - "Sync" device with iTunes
  - Open app in device

MAF Resources

- Oracle OTN
  http://www.oracle.com/technetwork/developer-tools/maf/overview/index.html
- Oracle MAF You Tube Channel
  http://www.youtube.com/user/OracleMobilePlatform
- Tutorials
  http://www.oracle.com/technetwork/developer-tools/maf/learnmore/index.html#tutorials
- Free Online Training Video
  http://download.oracle.com/otn_hosted_doc/maf/academy/DevelopingApplicationsWithMAF/presentation.html

King Training Resources

Wrapping it Up

- Oracle Mobile Application Framework (MAF) is a toolset you can use today to build and deploy hybrid mobile apps
- MAF's tooling insulates you against changes to standard libraries
- Creating MAF applications is relatively simple; building quality mobile apps will take some real design

Please Complete Session Evaluations

Build a Mobile App in 60 Minutes with MAF

To contact the author:

John King

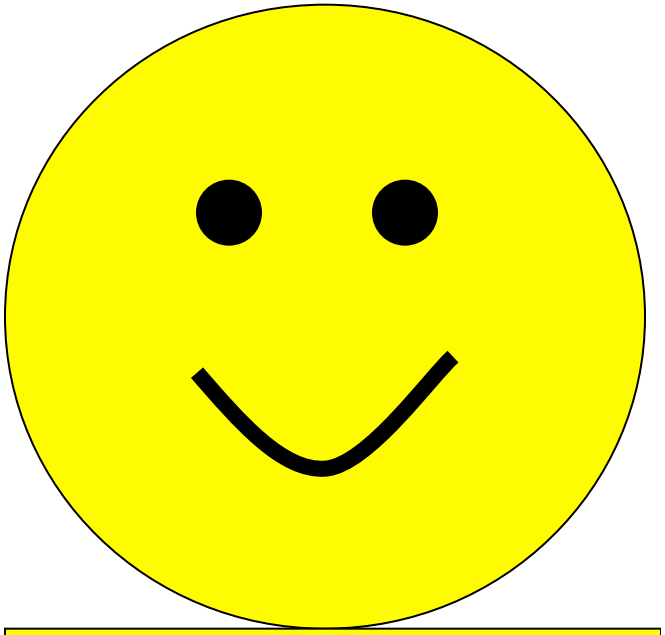
King Training Resources

P. O. Box 1780

Scottsdale, AZ 85252 USA

1.800.252.0652 - 1.303.798.5727

Email: john@kingtraining.com



Thanks for your attention!

Remember code OAUG17 for 10% discount before Jan 2018

Today's slides and examples are on the web:

http://www.kingtraining.com

- End