Real World Development using OpenEdge Mobile – some advanced features

Brian C. Preece
Ypsilon Software Ltd
brianp@ypsilonsoftware.co.uk

Purpose of this session

- To show how I used some advanced features of OpenEdge Mobile to develop a real-world shopping application
- Target audience: experienced OpenEdge developers
- Ideally you should have attended an introductory workshop or Brown Bag session on OpenEdge Mobile

Who are Ypsilon Software Ltd?

- Independent software development consultants
  - Consultancy
  - Bespoke Development
  - UI Design
  - Vendor selection
  - Training
- Main speciality: Progress Software tools and related subjects
- Principal consultant: Brian Preece
  - Developer since punched cards and paper tape!
  - Frequent speaker at Progress conferences on latest topics
  - Chairman of PUG UK and Ireland
  - Current focus: OpenEdge Mobile

Agenda

- Customer self scanning in retail stores
- Demo
- Using OpenEdge Mobile for mobile apps for business
- Code view
  - Recap of the basics
  - Using grids and properties to enhance appearance
  - Using “invoke” methods
  - The Appery.io API
  - Lighting up a grid line when it’s touched
  - Incremental development – adding services and methods to your app
- Important lessons/ issues/ bugs
- Conclusions and questions

Agenda

- Customer self scanning in retail stores
- Demo
- Using OpenEdge Mobile for mobile apps for business
- Code view
  - Recap of the basics
  - Using grids and properties to enhance appearance
  - Using “invoke” methods
  - The Appery.io API
  - Lighting up a grid line when it’s touched
  - Incremental development – adding services and methods to your app
- Important lessons/ issues/ bugs
- Conclusions and questions

Self scanning

- Customers scan their shopping as they go, using a dedicated device or mobile app
- They pack their shopping bags as they go round the store
- They check out through a dedicated checkout, avoiding the queues
- At random intervals, baskets are rescanned to discourage pilfering

Customers and stores both benefit

Benefit	Customer	Store
Fast checkout at special lane – minimal queuing	☺	☺
You can pack as you shop	☺	☺
Other customers and cashier can't see your purchases	☺	
Your basket total is calculated as you shop	☺	
You can be alerted to special offers and deals	☺	☺
The store gets detailed basket analysis by customer	?	☺

Why have an app for this?

- Saves money for the store
- Allows the app to extend beyond the store e.g. shopping list

The ultimate BYOD!

Agenda

- Customer self scanning in retail stores
- Demo
- Using OpenEdge Mobile for mobile apps for business
- Code view
  - Recap of the basics
  - Using grids and properties to enhance appearance
  - Using “invoke” methods
  - The Appery.io API
  - Lighting up a grid line when it’s touched
  - Incremental development – adding services and methods to your app
- Important lessons/ issues/ bugs
- Conclusions and questions

Self scanning Demonstration

Agenda

- Customer self scanning in retail stores
- Demo
- Using OpenEdge Mobile for mobile apps for business
- Code view
  - Recap of the basics
  - Using grids and properties to enhance appearance
  - Using “invoke” methods
  - The Appery.io API
  - Lighting up a grid line when it’s touched
  - Incremental development – adding services and methods to your app
- Important lessons/ issues/ bugs
- Conclusions and questions

What are hybrid mobile apps?

- iPhone launched in 2007, App Store in 2008
- First Android phone launched in 2008, Play Store in 2008 (as Android Market)
- iPhone apps programmed in Objective C, Android in Java
  - So apps had to be developed separately for both platforms
- HTML5/ CSS launched in 2012 to allow Responsive Web Apps to be developed
  - Particularly for small form factor of mobile devices
- PhoneGap Build launched 2012
  - Converted mobile web apps into native apps for Android and iOS, later Windows Phone
- Thus was born the Hybrid App
  - Not as tailored for the device as a native app
  - But could interact with device hardware
  - One app would work on any supported platform

What is OpenEdge Mobile?

- A toolset for creating hybrid apps
- Easy interface between a Progress AppServer and the Mobile App
- Visual Designer using JQuery Mobile UI components (Mobile App Builder)
  - Customised version of Appery.io by Exadel Corp.
  - Easy integration between Progress Developer Studio for OpenEdge (PDSOE) and Mobile AppBuilder (MAB)
  - Packages REST services into JavaScript Data Objects (JSDO)
  - Includes a subscription to PhoneGap Build
  - Includes a customised Tomcat web server
- Good points: easy Visual Designer, excellent integration with back end systems, uses many industry standard skills
- Bad points:
  - Very complex technology stack: OpenEdge ABL, REST, JSON, Tomcat, HTML5, CSS, JavaScript, Appery.io, JQuery, JQuery Mobile. (but this is the norm)
  - Hate JavaScript! – see later slide

What is RollBase Mobile?

- Extension to RollBase that allows hybrid mobile apps to be created!
- The Visual Designer is already there and looks much the same as the MAB of OE Mobile
- The Developer Studio part will be released with OpenEdge 11.4
- Most of what is in this presentation won’t change
- However hopefully the remaining bugs will be fixed!

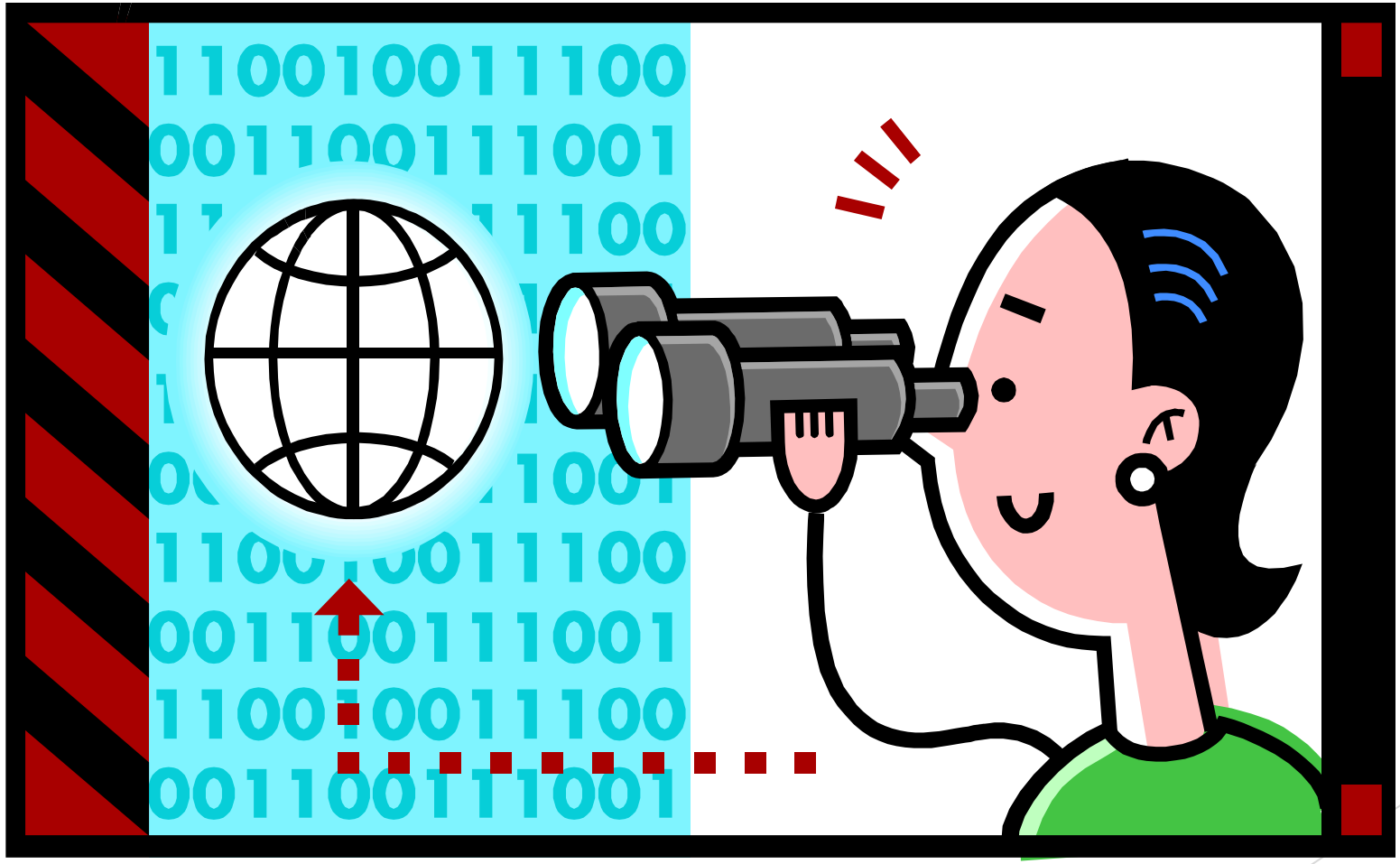
JavaScript vs ABL

- JavaScript has:
  - Java-like syntax
  - No proper OO
  - No type checking for variables or method signatures
  - No syntax checking
  - No complex data structures like collections
  - Variables don't have to be declared ***
- JQuery issues:
  - Many functions access DOM objects using quoted strings
- Conclusion: Perform as much business logic as possible in ABL

Agenda

- Customer self scanning in retail stores
- Demo
- Using OpenEdge Mobile for mobile apps for business
- Code view
  - Recap of the basics
  - Using “invoke” methods
  - Using grids and properties to enhance appearance
  - The Appery.io API
  - Lighting up a grid line when it’s touched
  - Incremental development – adding services and methods to your app
- Important lessons/ issues/ bugs
- Conclusions and questions

Code view



Agenda

- Customer self scanning in retail stores
- Demo
- Using OpenEdge Mobile for mobile apps for business
- Code view
  - Recap of the basics
  - Using grids and properties to enhance appearance
  - Using “invoke” methods
  - The Appery.io API
  - Lighting up a grid line when it’s touched
  - Incremental development – adding services and methods to your app
- Important lessons/ issues/ bugs
- Conclusions and questions

Some things to remember

- Services are invoked asynchronously
- Load event vs. Resume event
- Some things only work on the physical device
  - Not just hardware but also PhoneGap functions
- Map temp-table to grid not dataset
- Appery.io API is very basic, need JQuery for more complex tasks
- Remember to initialise all variables and empty temp-tables
- AppServer loads business entities from \wrk folder, so can't have two BE in different apps with the same name
- Don't change Nav Bar button text at run time just by changing the "text" property using the Appery.io API – need a JQuery command for this
- Consider not using CRUD business entities
- Back up your project using the Backup button in the MAB

Features/ bugs

- All keywords must be in upper case in Business Entities!
- Response parameters for “Invoke” methods have to be added manually
- Don’t base BE on temp-table using “like” syntax, use dataset instead or fully detailed temp-table definition

Agenda

- Customer self scanning in retail stores
- Demo
- Using OpenEdge Mobile for mobile apps for business
- Code view
  - Recap of the basics
  - Using grids and properties to enhance appearance
  - Using “invoke” methods
  - The Appery.io API
  - Lighting up a grid line when it’s touched
  - Incremental development – adding services and methods to your app
- Important lessons/ issues/ bugs
- Conclusions and questions

Agenda

- Customer self scanning in retail stores
- Demo
- Using OpenEdge Mobile for mobile apps for business
- Code view
  - Recap of the basics
  - Using grids and properties to enhance appearance
  - Using “invoke” methods
  - The Appery.io API
  - Lighting up a grid line when it’s touched
  - Incremental development – adding services and methods to your app
- Important lessons/ issues/ bugs
- Conclusions and questions

Conclusions

- OpenEdge Mobile is a great way to develop hybrid apps that have to connect to an OpenEdge system
- The Mobile AppBuilder provides a relatively easy method to develop the UI for your app and connect it to your AppServer
- Integration between the MAB and Dev Studio is excellent
- You will need some rudiments of JavaScript and JQuery to get the most out of OE Mobile
- My recommendation is to write as much business and UI logic as possible in ABL
- You can make use of third party resources to develop the look and feel of your app using CSS and HTML5
- This session only scratches the surface of what can be done with OE Mobile

Questions?

Thanks for your time!

brianp@ypsilonsoftware.co.uk

www.ypsilonsoftware.co.uk

Deploying on devices – Android is easy

- Two methods
  - Use Export button in MAB and download from QR code – doesn't always work
  - Use Android SDK
- SDK Method
  - Create .apk file using Export button in MAB or in Dev Studio, right click on the mobile app, select Copy Source Local, then Export Local
  - Install Android SDK from Google
  - Install device driver for your device
  - Ensure ADB utility recognises your device
  - Deploy with command line
    - ADB install –r "*full path/ filename of .apk file*"

Deploying on iOS (for testing)

- Preliminaries:
  - Get an Apple Developer Account
  - Register your device as a test device
  - Create certificate and provisioning file using this guide:
    - http://community.phonegap.com/nitobi/topics/detailed_guide_for_setting_up_building_ios_apps_without_a_mac (assuming you don't have a Mac)
  - Add the certificate and provisioning file onto the "iOS binary" page of App Settings
  - Make sure the Bundle ID on the iOS binary page matches that used to generate the provisioning file
  - Save the project
  - Compile the binary either using the Export button in the MAB or the right click options in Dev Studio

Ctd.

Deploying on iOS (ctd.)

- Download from the QR code generated by the Export button in the MAB
- Or:
  - In iTunes, select the Apps page.
  - Drag the .iap file into the Apps page
  - Your app is now available to be installed on your device
  - Connect your device to iTunes
  - Click on the Install button for your app in the apps page of your device
  - Click on “Apply” or “Sync”
  - Your app should now install
  - You must remove the app before you can install it again