

The IBM i Goes Mobile with Android

Joe Pluta
Pluta Brothers Design, Inc.



Agenda

- • **What's a Droid Anyway?**
- **Why Droid?**
 - And what about the iPhone?
- **Droid and the IBM i**
- **Conclusion**

What's a Droid, Anyway?

No, not that
kind of Droid!



**“These aren’t the
droids you’re
looking for.”**

- Obiwan

THIS Kind of Droid

Droid Incredible



Droid X



The Droid is a Smartphone

- **Droid is the Motorola branding of mobile devices using the Android operating system**
- **It's the generic term, like xerox or kleenex**
- **Other devices exist**
 - Big, little, in between
 - Cameras, wifi, bluetooth
 - HDMI
 - With or without keyboards
 - Even a front-facing keyboard - the Droid Pro is designed to be a Blackberry killer

Droid TV on Your HTC One



1.5 GHz Quad-core, 4.7" 1280x720 screen

More Android Goodness!



Samsung Galaxy Tab 10.1

Asus Eee Pad Transformer Prime



Agenda

- **What's a Droid Anyway?**

- • **Why Droid?**
 - And what about the iPhone?

- **Droid and the IBM i**

- **Conclusion**

Why Droid?

- **Availability/broad base**
- **Great features**
- **Active community**
- **Development tools**

Every Carrier has One

- **Verizon (many)**
- **T-Mobile**
- **Sprint**
- **US Cellular**
- **Even AT&T!**

Most Manufacturers as Well

- Motorola (of course)
- Samsung
- HTC
- LG
- Apple... **NOT!**

The Common Factor: Android

- **The Android OS is Google's mobile device operating system**
 - Technically, the OS was bought by Google
- **Android is based on Linux**
- **And Android's primary programming language is Java (the importance of which will become very clear)**

Android is Very Actively Developed

- **April 2009 - Version 1.5**
- **September 2009 - Version 1.6**
- **October 2009 - Version 2.0/2.1**
- **May 2010 - Version 2.2 (Froyo)**
- **December 2010 - Version 2.3 (Gingerbread)**
- **February 2011 – Version 3.0 (Honeycomb)**
 - May 2011 – Version 3.1
 - July 2011 – Version 3.2
- **October 2011 – Version 4.0 (Ice Cream Sandwich!)**
- **Sometime in 2012 – Version 5.0 (Jelly Bean)**

Android has Awesome Features

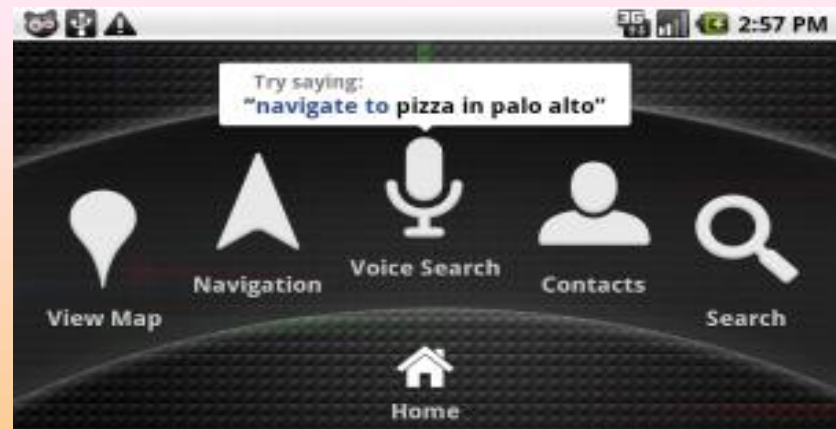
- **USB Tethering**
- **Flash 11.1**
- **Bluetooth 4.0**
- **WiFi Direct**
- **Multi-touch**
- **GPS**
- **Speech recognition**
- **Near Field Communication (NFC)**

Android Beam

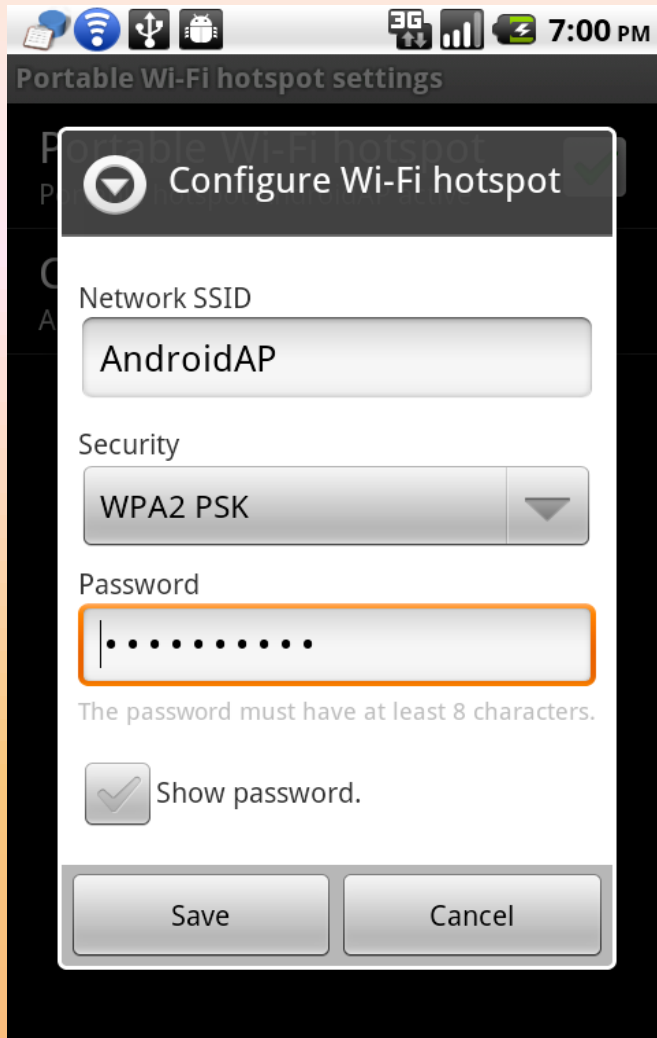
Android Beam allows any two NFC-enabled devices to talk to one another. You just tap the two devices together and they set up an interlink which you can then use to share just about anything between the two devices.

A whole new way to share contact info, eh?

Some Screenshots



More Shots



A Dynamic Evolving Environment

- **The features go on and on**
- **Every day new features are added, things you wouldn't even expect**
 - The network-based printer share application is awesome
 - A free barcode scanning API - just too cool
 - Voice and video chat
 - USB device support
 - Hot spot with HTTP proxy
- **So what drives this incredible innovation?**

Enormous Community

- **The Android development community is simply enormous**
- **Open source**
- **Thousands of applications**
- **And unlike certain fruit-based alternatives, the OS is very developer friendly**

Here's Your Home Page

ANDROID

developers

Home

SDK

Dev Guide

Reference

Resources

Videos

Blog

Developer Announcements



Introducing **Google Play**: An integrated digital content destination where users buy and enjoy all of their favorite content in one place. It's the new destination for Android apps!

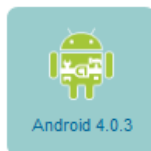
[Read more »](#)



Android Developers on Google+

We now have a Google+ page for [+Android Developers](#). We'll use it to host Hangouts for developers, talk about the latest releases, development and design tips, and much more.

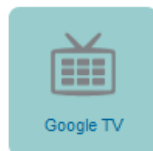
We're on 



Android 4.0.3



Google+ Page



Google TV



Download

The Android SDK has the tools, sample code, and docs you need to create great apps.

[Learn more »](#)



Publish

Google Play is an open service that lets you distribute your apps to devices.

[Learn more »](#)



Contribute

Android Open Source Project gives you access to the entire platform source.

[Learn more »](#)



Target Devices

The Device Dashboard provides information about deployed Android devices to help you target suitable device configurations as you build and update your apps.

[Learn more »](#)

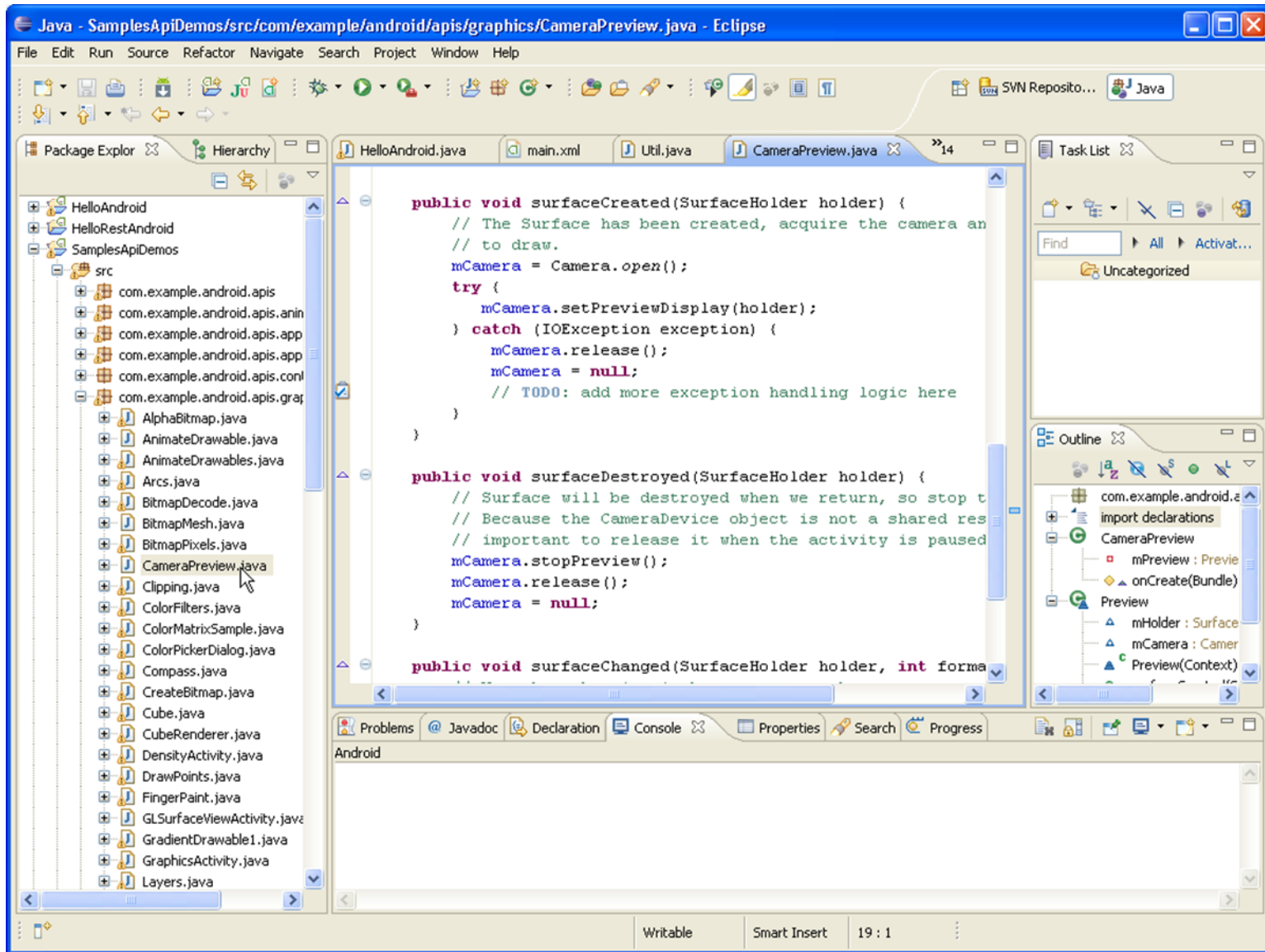
Development Tools

- Two words: Java and Eclipse
- The OS is Linux based, but the primary development language is Java
 - Android has its own JVM
 - Dalvik, named after a village in Eyjafjörður, Iceland
- If you know Java, you can program the Droid

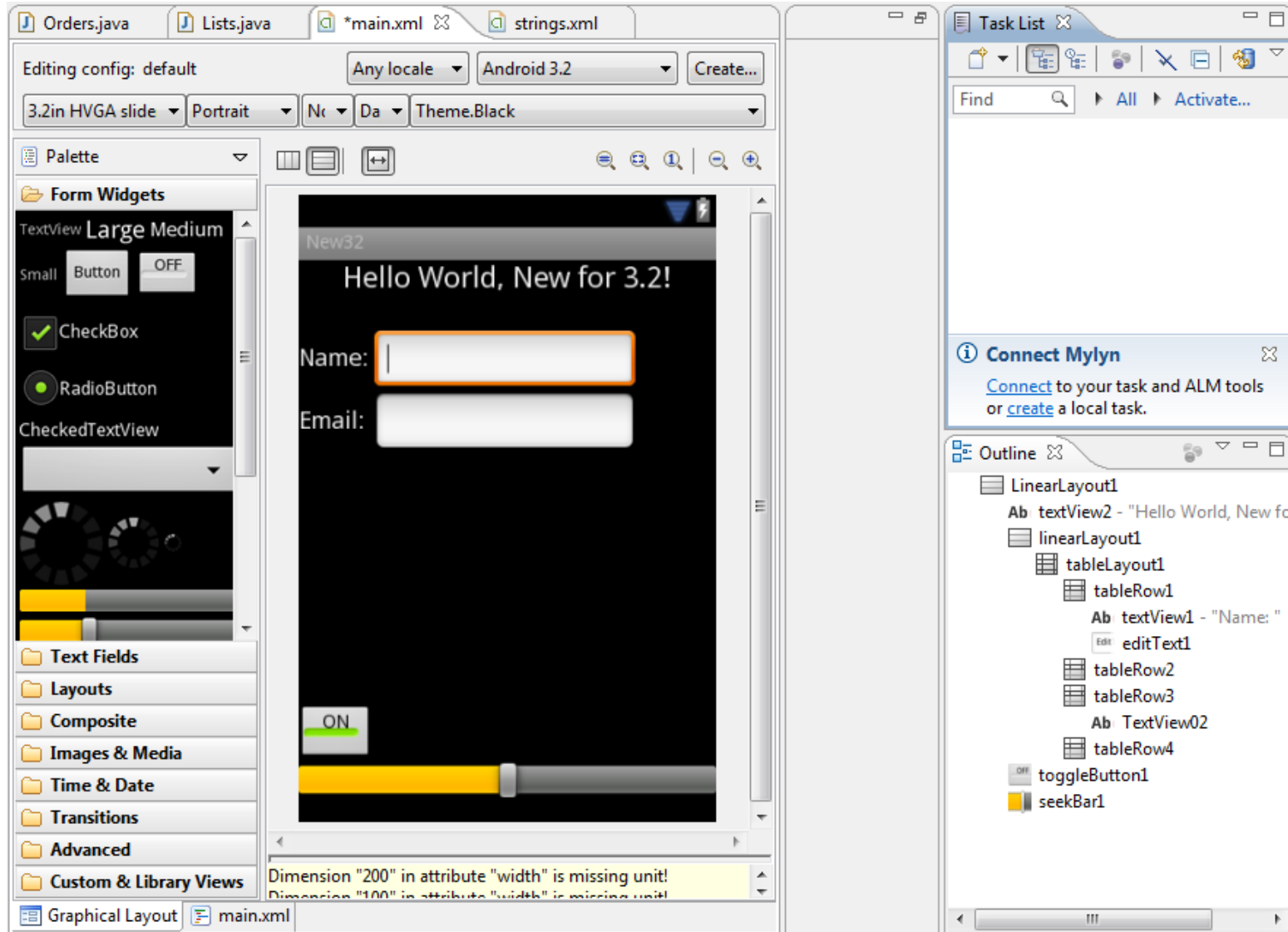
```
Toast.makeText(context, "Popup Message", Toast.LENGTH_LONG).show();
```

- That's why I say lots of IBM i programmers are already Android developers!

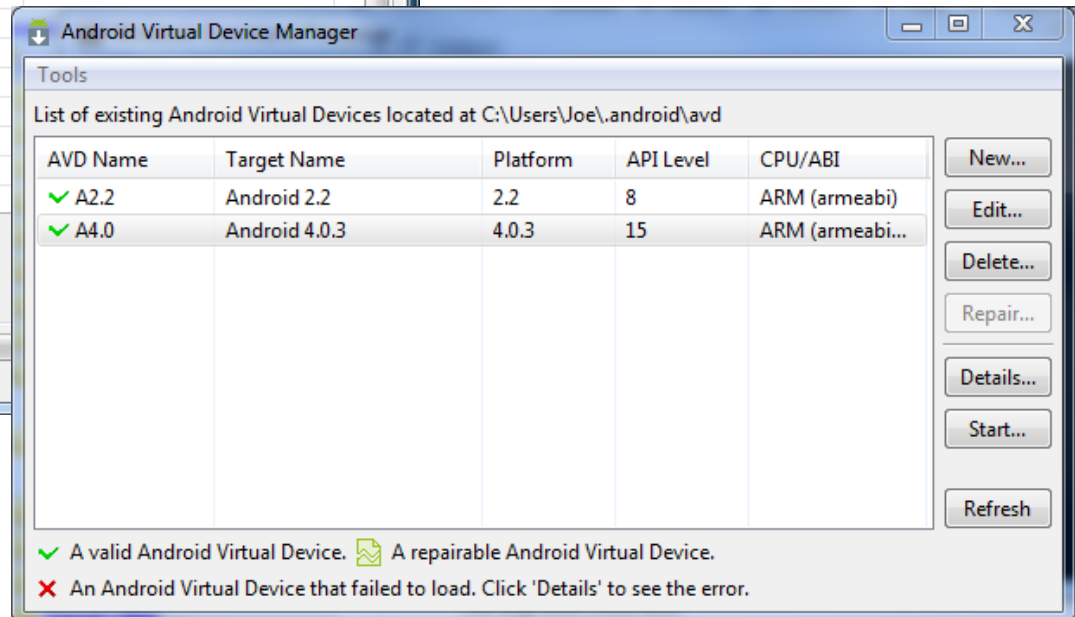
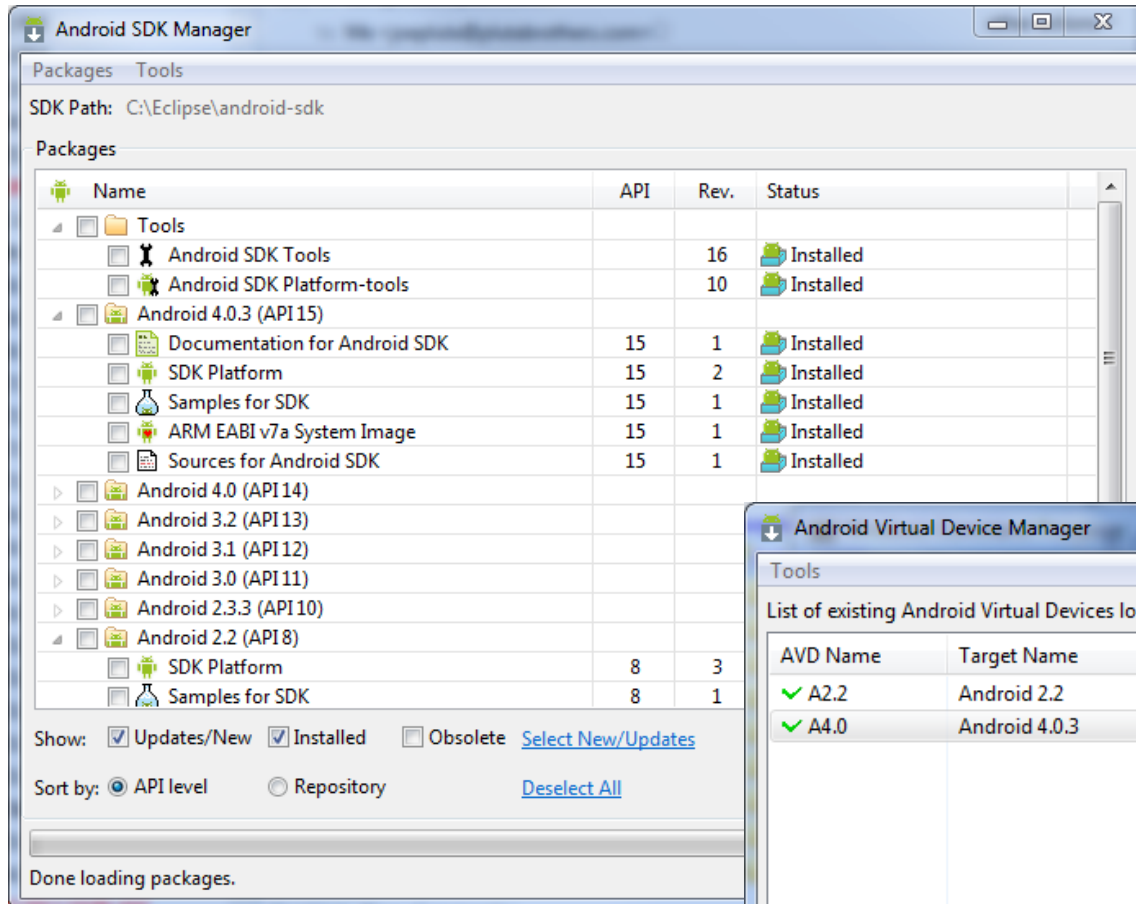
Android Has a Powerful Eclipse IDE



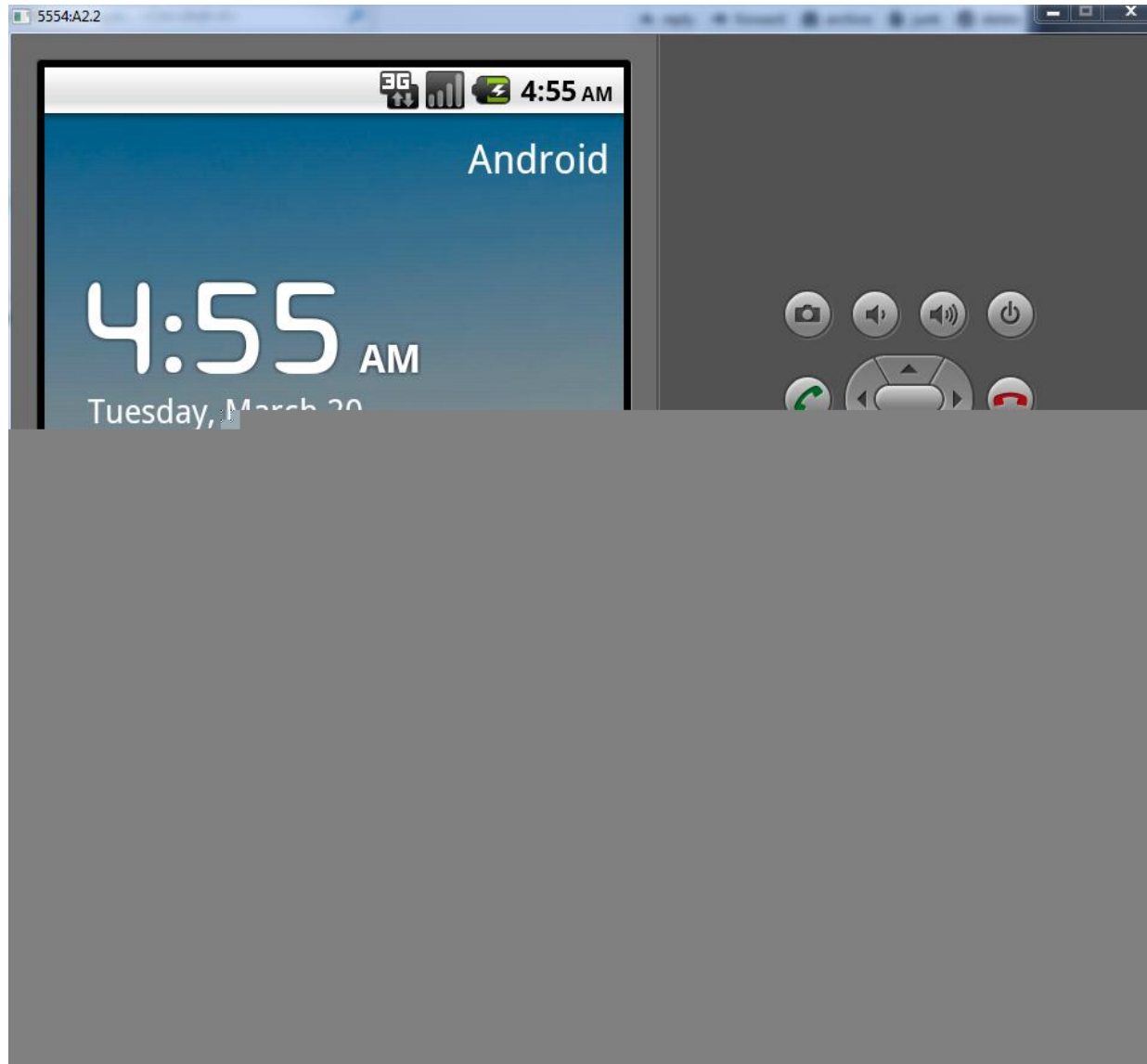
Including a WYSIWYG Designer



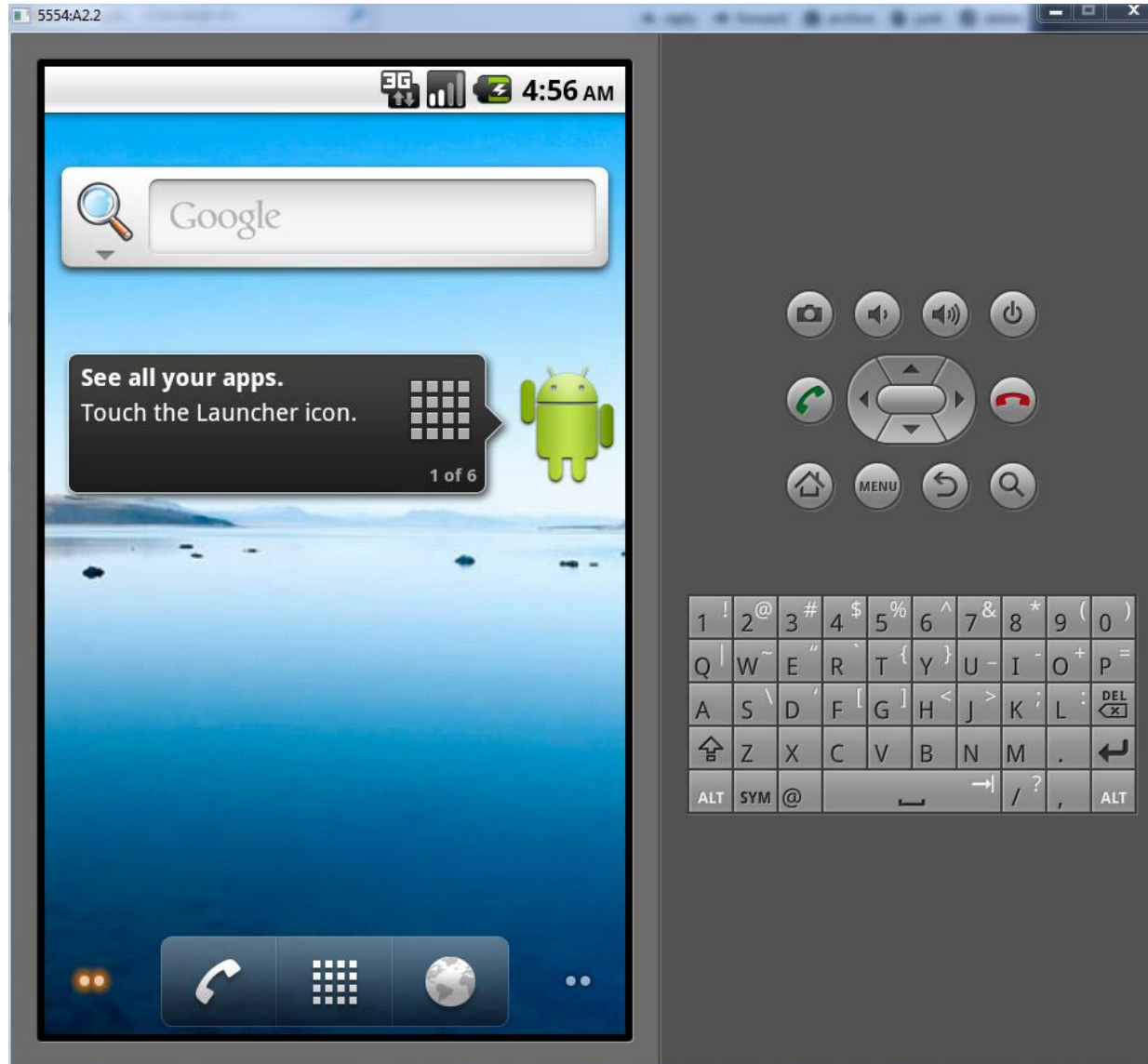
And a Full Multi-Version Emulator



Which Looks Like the Real Thing



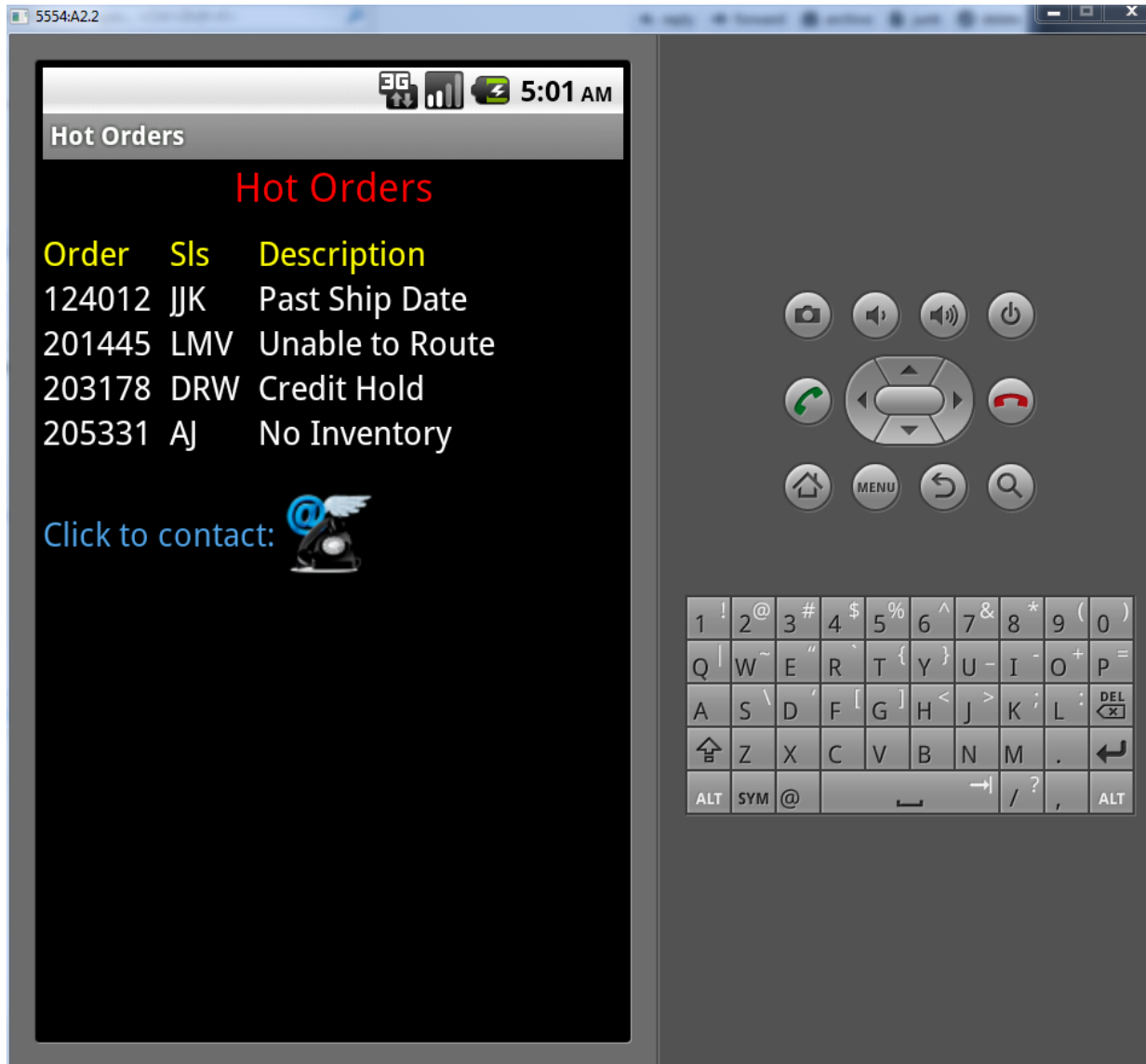
And Works Like the Real Thing



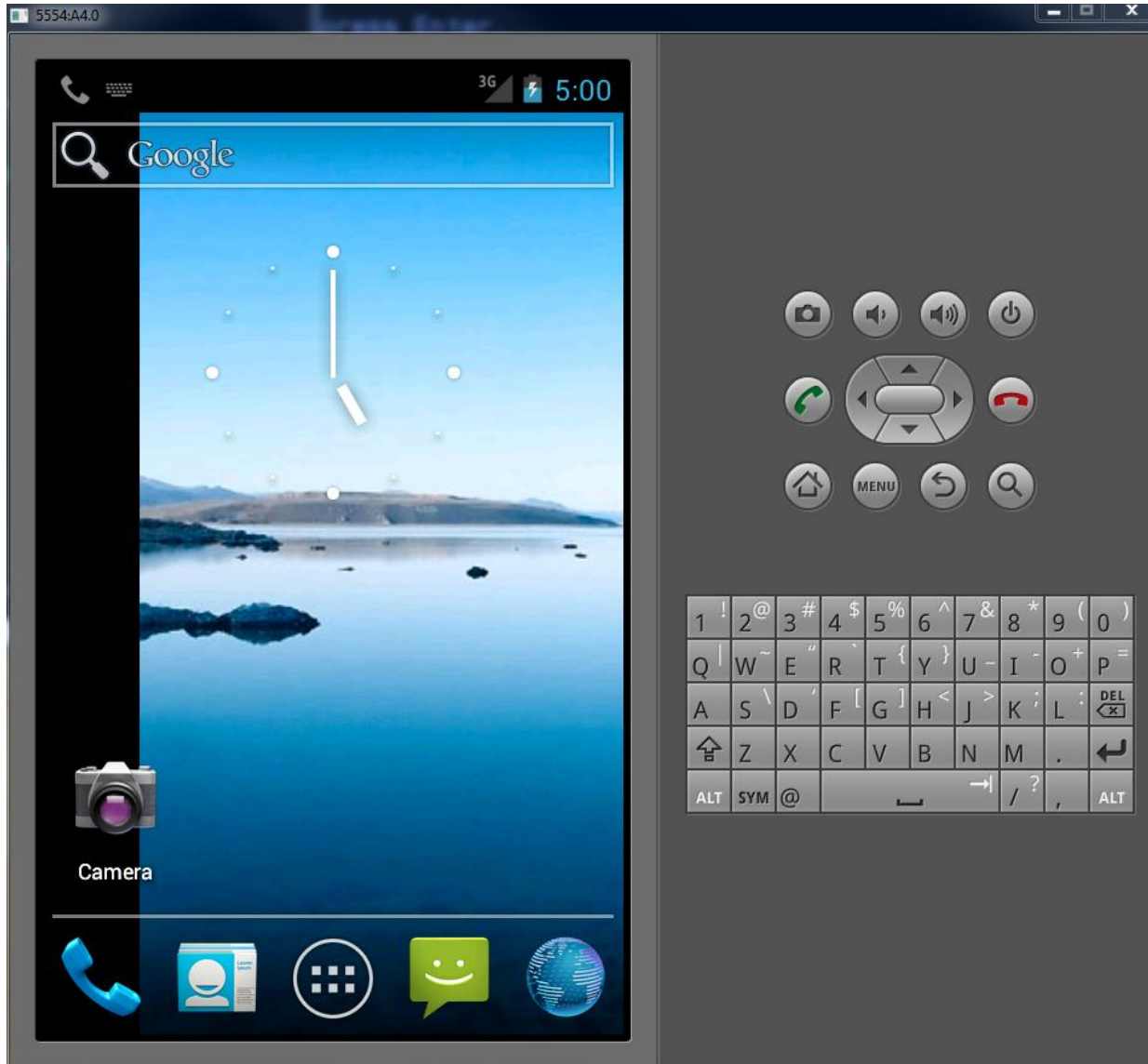
A Completely Realistic Experience



Android Apps on Your Desktop



Including Android 4.0



The iOS Platform

- Originally iPhone OS
- Extended to other Apple Products
- NOT licensed for any other hardware
- As of last summer, over half a million apps

iPhone OS

- Released as just OS X for the iPhone in June of 2007
- Originally, there was no plan for developing apps
- The idea was to build web apps that acted “just like native apps”
- That lasted until October, when the SDK was announced
 - And subsequently released to beta in March of 2008 and the OS was renamed iPhone OS
- In January 2010 the iPad was released and the following June Apple announced the rebranding of iPhone OS to iOS

iOS Extended to Other Platforms

- iPhone



- iPad



- iPod Touch



- Apple TV (iTV) – second generation

Not Licensed to 3rd-Party Hardware

- Even though it runs on standard x86 PC hardware
- Apple software licensing does not allow Apple OS software to run on devices that are not “Apple branded”
 - And they have the lawsuit to prove it
- There are ways around that, although the licensing issues are a bit murky for production applications
 - Search the word “Hackintosh”
 - Alternately, look for “OSx86”



The AppStore

- There's an app for that!



Android is Passing Apple in Numbers, if Not in Profit

For the first time Android app downloads are going to surpass those on Apple iPhones, analyst firm Ovum has predicted. Before the end of the year, Android will take a “significant lead,” with 8.1 billion downloads to Apple’s 6 billion, Ovum said.

FACTOID

For every \$1.00 in revenue for an iPhone app, the corresponding Android app makes only \$0.24.

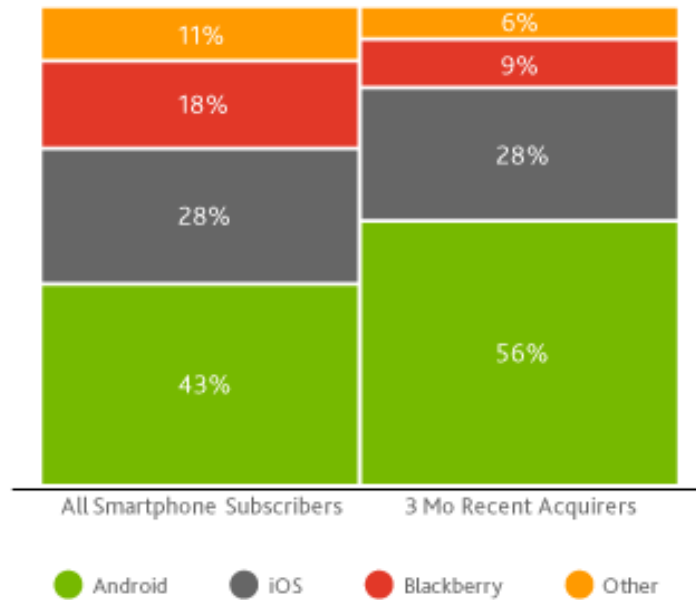
“ Although Android phones will lead on total download numbers, iPhone will continue to dominate the market in terms of revenues from paid-for apps. ”

Market Share, August 2011

In August, 56% of recent acquirers chose an Android device

Operating System Share – All Subscribers and Recent Acquirers

Aug 2011, Nielsen Mobile Insights, National



Source: Nielsen

nielsen

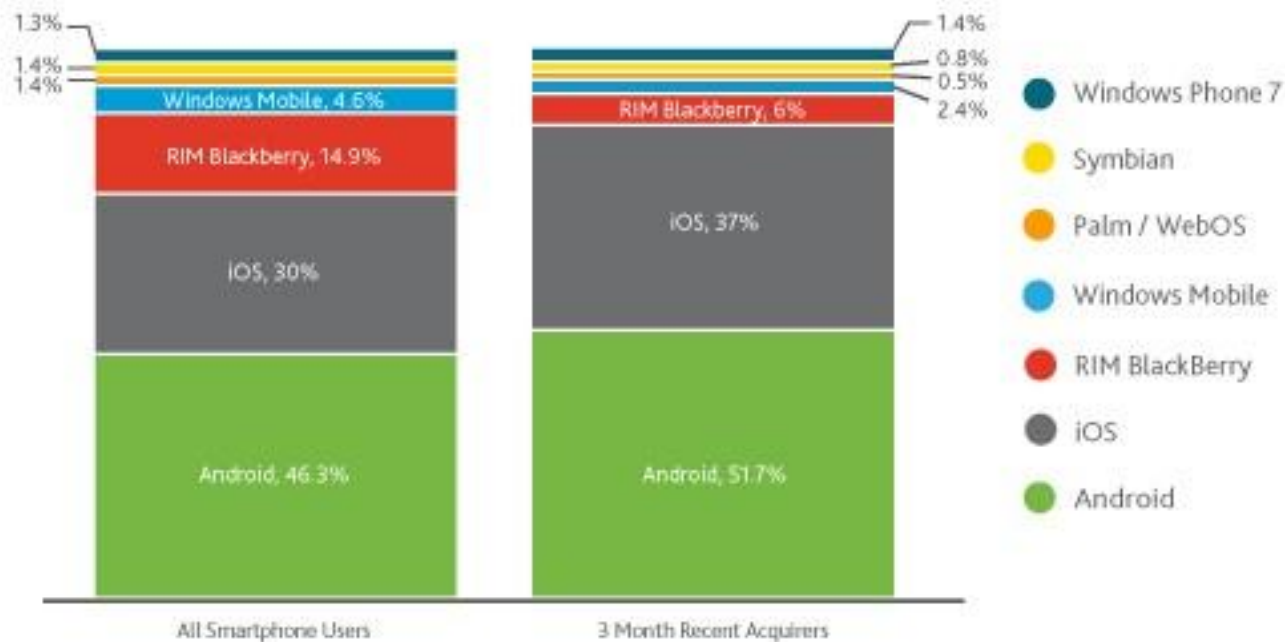
<http://techcrunch.com/2011/09/26/android-vs-iphone-sales/>

Market Share, Q4 2011

Android and iOS Squeezing Everyone Else

Operating System Share –
All Smartphone Consumers vs. Recent Smartphone Acquirers (3Mo).

Q4 2011, Nielsen Mobile Insights



Source: Nielsen

nielsen

Hello World in Objective-C

```
#import <Cocoa/Cocoa.h>
```

```
@interface World:NSObject  
{  
}  
-world;  
@end
```

Identifies the class World with a method world.

```
@implementation World  
- world  
{  
    NSLog(@"Hello World!");  
    return 0;  
}  
@end
```

Provides the implementation of the hello method.

```
int main(int argc, char *argv[])  
{  
    id hello;  
  
    hello=[[World alloc] init];  
    [hello world];  
  
    return 0;  
}
```

And this is your traditional main() method that invokes the World class.

Is Objective-C a Logical Next Step?

- For Objective-C you have to learn the basic C syntax, then you have to learn OOP concepts, and then finally you have to learn the Smalltalk message passing syntax (yes, there is dot notation as well, but it's not exactly prevalent)
- With Java you just need the C syntax and OOP; Java managed to integrate most of its OOP using standard C language constructs
 - Other than dot notation and the angle brackets for generics
- So I would steer you towards Java rather than Objective-C, and that's before taking into account the target market: Objective-C is iPads and iPhones, while Java is everywhere

Unless You're Absolutely Going Mac

- **Of course, if you're set on programming for the iPad**
- **And you're not going to use a browser-based solution**
- **Then you're definitely going to want to learn Objective-C**

Agenda

- **What's a Droid Anyway?**
- **Why Droid?**
 - And what about the iPhone?
- • **Droid and the IBM i**
- **Conclusion**

So It's a Great Environment

- That explains why you want Droid as your smartphone OS
- But why program for mobile devices at all?
- What can a smartphone do for my business applications?

A WHOLE LOT

Let's Start Easy

- **First off, even without any programming, the Droid is a powerful web-based UI device**
- **Any Froyo-level device supports both HTML5 and Flash 10.1**
 - So pretty much any web application you can imagine can run on the device
 - **Provided you can live within the real estate constraints**
 - **And you stay away from proprietary junk**
 - cough ... silverlight ... cough

Can't Any Mobile Device Do That ?

- **Sort of... the iGadgets can all handle HTML5 (Safari is a darned good browser)**
- **And they'll handle Flash someday**
 - Maybe, although that may now be a moot point
- **Windows Mobile devices are...**
 - Well, they're Windows
 - What can I say?

Why Does a Droid Help My Business?

- **A very portable browser can be of some help**
- **But that's really not enough to make it a true enterprise user interface**
 - Certainly not in the applications where a green screen makes sense
- **And that's where the thick client abilities of the Droid can really shine**

Let's Start With Some Capabilities

- **WiFi Connectivity**
- **Good resolution graphics**
- **Speech to text**
- **Image capture**
- **Bar code scanning**
- **Geo location**

WiFi Connectivity

- **This is a crucial component to mobile 5250 replacement**
 - This allows you to connect to the network without making the network publicly available
 - A traditional broadband access would require making the application accessible over the Internet
 - WiFi is more like having a VPN
 - And you can do other cool things like access network printers

Good Resolution Images

- **854 x 480 is nothing to sneeze at, especially for something that fits in your pocket**
- **It's equivalent to first generation notebook computers, and about a third of the pixels of a 1280x1024 monitor**
 - You may not want to play Blu-Ray on it, but it's perfectly acceptable for most applications

Speech to Text

- This is a feature with amazing potential
- The problem is that it's not particularly accurate yet, but that's supposed to be addressed in the next release (Jelly Bean)
- And the idea of being able to speak commands to the device has some serious appeal, especially in working environments

Text to Speech

- I left this off the list since it's kind of a related feature, but for hands-off processing, text to speech isn't bad either
- Imagine a completely hands-free application in which the application talks to you and you to it
- Especially appealing for road applications (see geo location later)

Image Capture

- **These phones have powerful cameras**
- **Most have at least a 5MP camera**
 - Many have 8MP cameras
 - And some support full 720p (HD) video capture!
- **It's not quite the same as a scanner, but it can do in a pinch**

Bar Code Scanning

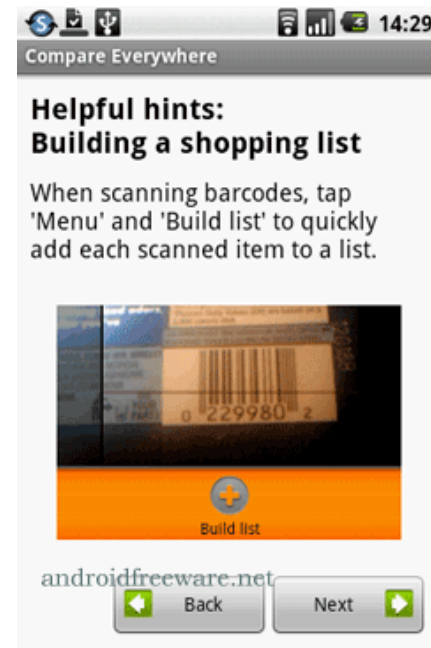
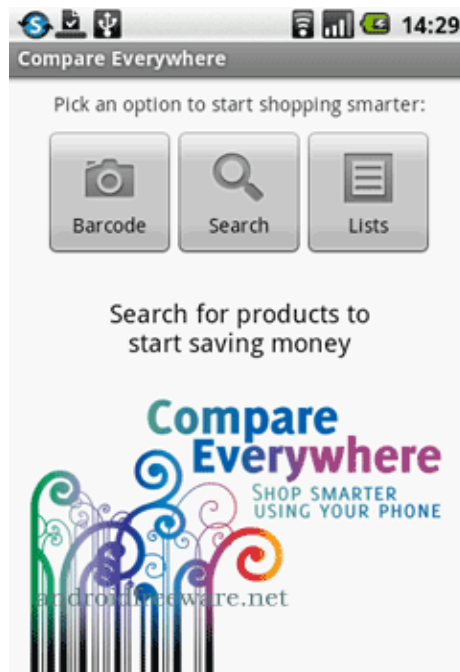
- **This they can do VERY well**
- **There are actually quite a few applications out there that support bar code scanning**
 - I'll talk about that in a moment
- **Open source libraries for bar code scanning are readily available**

Geo Location

- **Most devices support location information**
- **This allows you to determine the physical location of the device**
- **The most precise devices (using GPS) are typically accurate to 10 meters or so, so they can actually get you to a specific address**

Applications Already Out There

- There's a really powerful application called CompareEverywhere




Here are the Results

Compare Everywhere

**Helpful hints:
Mapping local stores**

When looking at a product, you can see all local stores on an overview map. Just tap 'Menu' then 'Map Stores'.




androidfreeware.net

Back Next

The screenshot shows a mobile application interface. At the top, there's a status bar with icons for signal, battery, and time (14:28). Below that is a header 'Compare Everywhere'. The main content area has a title 'Helpful hints: Mapping local stores' followed by a paragraph explaining that users can see local stores on a map by tapping 'Menu' then 'Map Stores'. Below the text is a map showing three store locations with callout boxes: Barnes and Noble at \$9.99, Wal-Mart at \$8.88, and Best Buy at \$9.99. At the bottom, there's a watermark 'androidfreeware.net' and two buttons labeled 'Back' and 'Next' with arrow icons.

Compare Everywhere



Hello, Android
[Paperback]
Burnett

Online stores

Buy.com	\$19.99
Amazon.com	\$21.75
Deepdiscount.com	\$28.66
buyALLbooks.com	\$29.00
androidfreeware.net Eruditor.com	\$29.29

The screenshot shows the same mobile application interface. The status bar shows the time as 14:31. The header is 'Compare Everywhere'. Below the header is a book cover for 'Hello, Android' by Burnett. To the right of the cover, the text reads 'Hello, Android [Paperback] Burnett'. Below this is a section titled 'Online stores' which contains a list of retailers and their prices. The list includes Buy.com (\$19.99), Amazon.com (\$21.75), Deepdiscount.com (\$28.66), buyALLbooks.com (\$29.00), and androidfreeware.net/Eruditor.com (\$29.29).

How Does This Work with the IBM i?

- **For web applications, there's literally no other programming involved**
- **Any existing applications will work just fine with the Droid**
 - Within the caveats of the screen size

But if You Can Code a Little Java

- **You can write thick client applications**
- **The easiest method to communicate with the IBM i is HTTP**
 - Write SOAP or REST services
 - Access them using an HTTP session object
 - Fully supported and easy to implement
- **These applications can be private, using only WiFi and the local LAN, or public using the Internet**

The RPG Side of Things

- **Use EGL to write SOAP services calling RPG**
 - Simple, very easy to write
 - Allows you to expose those same services to other applications
- **Or you write out JSON strings**
 - These are relatively easy to format
 - They can be easily digested by Android APIs
 - Lots of tools, like RPG OA or CGIDEV

Development Cycle

- The really nice thing is that you can do all the development and testing right on your desktop
- In fact, because the SDK plugs into Eclipse, you can integrate it into either an RDPI or RDi-SOA workbench
- Try doing that with any other multi-platform environment!

What's the Business Benefit?

- That depends on what you're trying to accomplish
- Several different possibilities come to mind immediately

Application 1

- **On the dock**

- Imagine an application where you can use your phone to record truck loading
- Scan the bar code for each pallet or container as it is loaded
- Optionally take photographs and store them
- Scan the PRO number
- Scan the BOL number
- Scan the printed BOL and/or signature

Signature capture capability is getting better...

Application 1

- **The interesting thing about this application is that you could theoretically make it available for the truck drivers**
 - They could load their own trucks, scan the pallets themselves, and do everything else right up to printing the BOL, signing it and scanning it
 - Obviously, it depends on your relationship with the carrier, but it has some interesting potential

Application 2

- **Inventory / Picking**
 - Another use of the bar code scanner
 - The picker could use a barcoded pick list to determine what to pick
 - The locations could be presented on a map of the warehouse
 - You could display images to help verify the item that was being picked
- **Variations of this could be used for picking, cycle counting, you name it**
- **No more 5250 terminals mounted on forklifts!**

Application 3

- **Hands-free interface**

- Say you're doing a visual inspection in a difficult environment
 - **Gloves required**
- You can use a bluetooth headset and then use speech-to-text and text-to-speech to identify and record issues
- You can also record position information with geo location

- **This doesn't necessarily require thick client**

- It can be done in the web, but it's nice that you can use the same device for thick and thin apps

Agenda

- **What's a Droid Anyway?**
- **Why Droid?**
 - And what about the iPhone?
- **Droid and the IBM i**
- • **Conclusion**

7 Key Points to Take Home

- **The mobile device may be your next most important user interface**
 - On the dock, in the airport, at the hotel
- **Android is the best smartphone solution**
 - iPhones are cool, but Droids get the job done
 - Apple's antipathy to the outside world is legendary
 - Who wants to write in Objective-C?
- **And Android's not just for phones – it's tablets, too!**
 - Move over iPad, hello Galaxy Tab!

7 Key Points to Take Home

- **Droid devices are available to fit nearly every niche**
 - Big screen? Light weight? Keyboard? Camera?
 - You name it, there's a Droid for you
- **Application development can be done to fit your need**
 - Thin/rich client? Got you covered.
 - *HTML5 and Flash 11*
 - Thick client? All over that as well.
 - *Particularly good for applications that access networks or the device's camera*

7 Key Points to Take Home

- **The development tools are first rate**
 - ◆ Tools for development and testing on the desktop and external deployment are outstanding
 - ◆ And you get to code in Java!
 - Yet another reason to make Java your primary external language
- **Deployment is simple**
 - ◆ If you want to deploy an application publicly, you can use Android market
 - ◆ Or deploy locally using a network or an SD card

Think of it this way...

Mobile devices are the next evolutionary step: they make thick client programming make sense again.

Resources

- <http://www.android.com>
- http://en.wikipedia.org/wiki/Comparison_of_Android_devices
- **Professional Android 2 Application Development (Wrox Programmer to Programmer)**
- <http://www.androidcentral.com/>