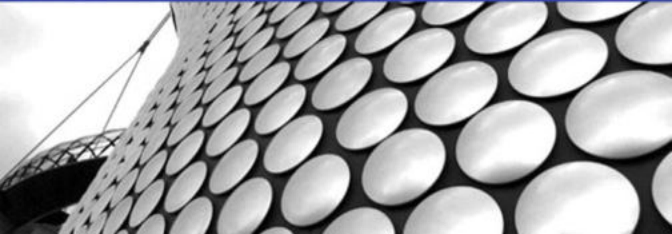
James Steele
Nelson To

The Android Developer's Cookbook

Building Applications with the Android SDK

Developer's Library



The Android Developer's Cookbook

Building Applications with the Android SDK

The Android Developer's Cookbook
Building Applications with the Android SDK

James Steele
Nelson To

Addison-Wesley

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Cape Town • Sydney • Tokyo • Singapore • Mexico City

Library of Congress Cataloging-in-Publication Data

Steele, James, 1971-
The Android developer's cookbook : building applications with the Android SDK / James Steele, Nelson To.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-0-321-74123-3 (pbk. : alk. paper)
ISBN-10: 0-321-74123-4 (pbk. : alk. paper)
1. Application software—Development. 2. Android (Electronic resource) 3. Mobile computing. 4. Smartphones—Programming. 5. Operating systems (Computers) I. To, Nelson, 1976- II. Title.
QA76.76.A65S743 2011
004.1675—dc22
2010033254

Text printed in the United States on recycled paper at RR Donnelley, Crawfordsville, Indiana.

First Printing: October 2010

ISBN-10: 0-321-74123-4
ISBN-13: 978-0-321-74123-3

Editor-in-Chief
Mark Taub

Acquisitions Editor
Trina McDonald

Development Editor
Michael Thurston

Managing Editor
Sandra Schroeder

Project Editor
Mandie Frank

Copy Editor
Deadline Driven Publishing

Indexer
Erika Millen

Proofreader
Jovana Shirley

Technical Editors
Romin Irani
Douglas Jones

Publishing Coordinator
Olivia Basegio

Designer
Gary Adair

Page Layout
Mark Shirar

❖

To Wei with love.

Jim

To my dear mom.

Nelson

❖

Contents at a Glance

1 Overview of Android 1

2 Application Basics: Activities and Intents 23

3 Threads, Services, Receivers, and Alerts 51

4 User Interface Layout 79

5 User Interface Events 117

6 Multimedia Techniques 147

7 Hardware Interface 169

8 Networking 195

9 Data Storage Methods 221

10 Location-Based Services 251

11 Advanced Android Development 277

12 Debugging 303

Index 317

l......

Table of Contents

1 Overview of Android 1
The Evolution of Android 1
The Dichotomy of Android 2
Devices Running Android 2
HTC Models 6
Motorola Models 6
Samsung Models 6
Tablets 7
Other Devices 7
Hardware Differences on Android Devices 8
Screens 8
User Input Methods 9
Sensors 9
Features of Android 10
Multiprocess and App Widgets 11
Touch, Gestures, and Multitouch 11
Hard and Soft Keyboards 11
Android Development 11
How to Use the Recipes in This Book 12
Designing Applications Well 12
Maintaining Forward Compatibility 13
Robustness 13
Software Development Kit 14
Installing and Upgrading 14
Software Features and API Level 15
Emulator and Android Device Debug 16
Using the Android Debug Bridge 18
Signing and Publishing 18
Android Market 19
End-User License Agreement 19
Improving App Visibility 19
Differentiating an App 20
Charging for an App 20
Managing Reviews and Updates 21
Alternatives to the Android Market 22

2 Application Basics: Activities and Intents 23

Android Application Overview 23

Recipe: Creating a Project and an Activity 24

Directory Structure of Project and Autogenerated Content 26

Android Package and Manifest File 28

Renaming Parts of an Application 30

Activity Lifecycle 30

Recipe: Utilizing Other Lifecycle Functions 31

Recipe: Forcing Single Task Mode 33

Recipe: Forcing Screen Orientation 34

Recipe: Saving and Restoring Activity Information 34

Multiple Activities 35

Recipe: Using Buttons and TextView 36

Recipe: Launching Another Activity from an Event 37

Recipe: Launching an Activity for a Result Using Speech to Text 41

Recipe: Implementing a List of Choices 43

Recipe: Using Implicit Intents for Creating an Activity 44

Recipe: Passing Primitive Data Types Between Activities 46

3 Threads, Services, Receivers, and Alerts 51

Threads 51

Recipe: Launching a Secondary Thread 51

Recipe: Creating a Runnable Activity 55

Recipe: Setting a Thread’s Priority 57

Recipe: Canceling a Thread 57

Recipe: Sharing a Thread Between Two Applications 58

Messages Between Threads: Handlers 58

Recipe: Scheduling a Runnable Task from the Main Thread 59

Recipe: Using a Countdown Timer 61

Recipe: Handling a Time-Consuming Initialization 62

Services 64

Recipe: Creating a Self-Contained Service 65

Adding a Broadcast Receiver 69
Recipe: Starting a Service When the Camera Button Is Pressed 70
App Widgets 72
Recipe: Creating an App Widget 72
Alerts 74
Recipe: Using Toast to Show a Brief Message on the Screen 74
Recipe: Using an Alert Dialog Box 75
Recipe: Showing Notification in Status Bar 76

4 User Interface Layout 79
Resource Directories and General Attributes 79
Recipe: Specifying Alternate Resources 81
Views and ViewGroups 82
Recipe: Building Layouts in the Eclipse Editor 83
Recipe: Controlling the Width and Height of UI Elements 86
Recipe: Setting Relative Layout and Layout ID 89
Recipe: Declaring a Layout Programmatically 90
Recipe: Updating a Layout from a Separate Thread 92
Text Manipulation 94
Recipe: Setting and Changing Text Attributes 95
Recipe: Providing Text Entry 98
Recipe: Creating a Form 100
Other Widgets: From Buttons to Seek Bars 101
Recipe: Using Image Buttons in a Table Layout 102
Recipe: Using Check Boxes and Toggle Buttons 105
Recipe: Using Radio Buttons 108
Recipe: Creating a Drop-Down Menu 110
Recipe: Using a Progress Bar 112
Recipe: Using a SeekBar 114

5 User Interface Events 117
Event Handlers and Event Listeners 117
Recipe: Intercepting a Physical Key Press 117
Recipe: Building Menus 121

Recipe: Defining Menus in XML 126
Recipe: Utilizing the SEARCH Key 127
Recipe: Reacting to Touch Events 128
Recipe: Listening for Fling Gestures 130
Recipe: Using Multitouch 133
Advanced User Interface Libraries 136
Recipe: Using Gestures 136
Recipe: Drawing 3D Images 140

6 Multimedia Techniques 147
Images 148
Recipe: Loading an Image for Manipulation 148
Audio 154
Recipe: Choosing and Playing Back Audio Files 154
Recipe: Recording Audio Files 157
Recipe: Manipulating Raw Audio 158
Recipe: Using Sound Resources Efficiently 163
Recipe: Adding Media and Updating Paths 165
Video 165

7 Hardware Interface 169
Camera 169
Recipe: Customizing the Camera 170
Other Sensors 175
Recipe: Getting a Device’s Rotational Attitude 176
Recipe: Using the Temperature and Light Sensor 179
Telephony 180
Recipe: Utilizing the Telephony Manager 181
Recipe: Listening for Phone States 183
Recipe: Dialing a Phone Number 185
Bluetooth 185
Recipe: Turning on Bluetooth 186
Recipe: Discovering Bluetooth Devices 187
Recipe: Pairing with Bonded Bluetooth Devices 188
Recipe: Opening a Bluetooth Socket 188
Recipe: Using Device Vibration 191
Recipe: Accessing the Wireless Network 191

8 Networking 195

Using SMS 195

Recipe: Autosend an SMS Based on a Received SMS 197

Using Web Content 204

Recipe: Customizing a Web Browser 204

Recipe: Using an HTTP GET 204

Recipe: Using HTTP POST 209

Social Networking 210

Recipe: Integrating with Twitter 210

9 Data Storage Methods 221

Shared Preferences 221

Recipe: Creating and Retrieving Shared Preferences 222

Recipe: Using the Preferences Framework 222

Recipe: Changing the UI Based on Stored Data 225

Recipe: Adding a EULA 228

SQLite Database 232

Recipe: Creating a Separate Database Package 232

Recipe: Using a Separate Database Package 236

Recipe: Creating a Personal Diary 239

Content Provider 243

Recipe: Creating a Custom Content Provider 244

File Saving and Loading 249

10 Location-Based Services 251

Location Basics 251

Recipe: Retrieving Last Location 253

Recipe: Updating Location Upon Change 254

Recipe: Listing All Enabled Providers 256

Recipe: Translating a Location to Address (Reverse Geocoding) 258

Recipe: Translating an Address to Location (Geocoding) 261

Using Google Maps 263

Recipe: Adding Google Maps to an Application 265

Recipe: Adding Markers on a Map 267

Recipe: Adding Views to a Map 271
Recipe: Marking the Device's Current Location on a Map 274
Recipe: Setting up a Proximity Alert 274

11 Advanced Android Development 277
Android Custom View 277
Recipe: Customizing a Button 277
Android Native Components 283
Recipe: Developing a Native Component 284
Android Security 287
Recipe: Declaring and Enforcing Permissions 288
Android Inter-Process Communication 288
Recipe: Implementing a Remote Procedure Call 289
Android Backup Manager 294
Recipe: Creating a Backup of Runtime Data 294
Recipe: Backing Up Files to the Cloud 296
Recipe: Triggering Backup and Restore 296
Android Animation 298
Recipe: Creating an Animation 299

12 Debugging 303
Eclipse Built-in Debug Tools 303
Recipe: Specifying a Run Configuration 303
Recipe: Using the DDMS 304
Recipe: Debugging Through Breakpoints 306
Android SDK Debug Tools 307
Recipe: Using the Android Debug Bridge 307
Recipe: Using LogCat 307
Recipe: Using the Hierarchy Viewer 309
Recipe: Using TraceView 311
Android System Debug Tools 313
Recipe: Setting up GDB Debugging 315

Index 317

Preface

Android is the fastest growing mobile operating system (OS). With over 30 smartphones introduced in the last year and over 10,000 applications (apps) being added every month, the Android ecosystem is growing as well. There is enough diversity in device features and wireless carriers to appeal to just about anyone.

Netbooks have always been a natural platform to adopt Android, but the inertia behind Android has fed the growth further into televisions and even automobiles. Many of the world's largest corporations—from banks to fast food chains to airlines—ensure a presence in Android and offer compatible services. Android developers have many opportunities, and relevant apps reach more people than ever before, increasing the satisfaction of creating a relevant app.

Why an Android Cookbook?

The Android OS is simple to learn, and Google provides many libraries to make it easy to implement rich and complex applications. The only aspect lacking, as mentioned by many in the Android developer community, is clear and well-explained documentation. The fact that Android is open source means anyone can dive in and reverse engineer some documentation. Many developer bulletin boards have excellent examples deduced using exactly this method. Still, a book that has a consistent treatment across all areas of the OS is useful.

In addition, a clear working example is worth a thousand words of documentation. Developers faced with a problem usually prefer to do a form of extreme programming; that is, they find examples of working code that does something close to the solution and modify or extend it to meet their needs. The examples also serve as a way to see the coding style and help to shape other parts of the developer's code.

This Android Cookbook serves to fill a need by providing many various self-contained recipes. As each recipe is introduced, the main concepts of the Android OS are also explained.

Who Should Read This Book?

Users who are writing their own Android applications will get the most out of this cookbook. Basic familiarity with Java and the Eclipse development environment is assumed, but not required for the majority of the book. Java is a modular language and most (if not all) of the example recipes can be incorporated with minimal change to the reader's own Android project. The motivation for each topic lends itself well for use as an Android course supplement.

Utilizing Recipes

In general, the code recipes in this cookbook are self-contained and include all the information necessary to run a working application on an Android device. Chapters 1 and 2 give an introduction to the overall use of Android, but feel free to jump around and start using whatever is necessary.

This book is written first as a reference, providing knowledge mostly by example with greatest benefits through implementation of the recipes of interest. The main technique introduced in each recipe is specified in the section heading. However, additional techniques are included in each recipe as needed to support the main recipe.

After reading this book, a developer should

- Be able to write an Android Application from scratch.
- Be able to write code that works across multiple versions of Android.
- Be able to utilize the various Application Programming Interfaces (APIs) provided in Android.
- Have a large reference of code snippets to quickly assimilate into applications.
- Appreciate the various ways to do the same task in Android and the benefits of each.
- Understand the unique aspects of Android programming techniques.

Book Structure

Chapter 1, "Overview of Android," provides an introduction to all aspects of Android outside of the code itself. It is the only chapter that doesn't include recipes, but provides useful background material. Chapter 2, "Application Basics: Activities and Intents," provides an overview of the four Android components and explanation of how an Android project is organized. It also focuses on the activity as a main application building block. Chapter 3, "Threads, Services, Receivers, and Alerts," introduces background tasks such as threads, services, and receivers, as well as notification methods for these background tasks using alerts. Chapter 4, "User Interface Layout," covers the user interface screen layout and views, and Chapter 5, "User Interface Events," covers the user initiated events such as touch events and gestures.

Chapter 6, "Multimedia Techniques," covers multimedia manipulation and record and playback of audio and video. Chapter 7, "Hardware Interface," introduces the hardware APIs available on Android devices and how to utilize them. Chapter 8, "Networking," discusses interaction outside of the Android device with SMS, web browsing, and social networking. Chapter 9, "Data Storage Methods," covers various data storage techniques available in Android including SQLite. Chapter 10, "Location-Based Services," focuses on accessing the location through various methods such as GPS and utilizing services such as the Google Maps API. Chapter 11, "Advanced Android Development," provides some advanced techniques in Android including customizing views, using native code for faster processing, and utilizing the Android Backup Manager. Finally, Chapter 12, "Debugging," provides the testing and debugging framework useful throughout the development cycle.

Additional References

There are many online references for Android. A few essential ones are

- Android Source Code: http://source.android.com/
- Android Developer Pages: http://developer.android.com/
- Android Developer Forums: http://www.svcAndroid.com/
- Open Source Directory: http://osdir.com/
- Stack Overflow Discussion Threads: http://stackoverflow.com/
- Talk Android Developer Forums: http://www.talkandroid.com/android-forums/

About the Authors

James Steele was doing post-doctoral work in physics at MIT when he decided to join a startup in Silicon Valley. Fifteen years later and he continues to innovate, bringing research projects to production in both the consumer and mobile market. He actively presents and participates in various Silicon Valley new technology groups.

Nelson To has more than ten applications of his own in the Android Market. He also has worked on enterprise Android applications for Think Computer, Inc. (PayPhone), AOL (AIM), Stanford University (Education App), and Logitech (Google TV). He also assists in organizing the Silicon Valley Android Meetup Community and teaches Android classes both in the Bay Area and China.

1 Overview of Android

The Android operating system (OS) has come a long way since the announcement of the Open Handset Alliance in late 2007. The idea of an open source OS for embedded systems was not new, but Google aggressively backing it definitely has helped push Android to the forefront in just a few years.

Many wireless carriers in multiple countries across various communication protocols have one or more Android phones available. Other embedded devices, such as tablets, netbooks, televisions, set-top boxes, and even automobiles, have also adopted the Android OS.

This chapter discusses various general aspects of Android useful for a developer. It provides a foundation for the creation of Android applications and a context for the recipes in the rest of this book.

The Evolution of Android

Google, seeing a large growth of Internet use and search in mobile devices, acquired Android, Inc., in 2005 to focus its development on a mobile device platform. Apple introduced the iPhone in 2007 with some ground-breaking ideas including multitouch and an open market for applications. Android was quickly adapted to include these features and to offer definite distinctions, such as more control for developers and multitasking. In addition, Android incorporates enterprise requirements, such as exchange support, remote wipe, and Virtual Private Network (VPN) support, to go after the enterprise market that Research In Motion has developed and held so well with its Blackberry models.

Device diversity and quick adaptation have helped Android grow its user base, but it comes with potential challenges for developers. Applications need to support multiple screen sizes, resolution ratios, keyboards, hardware sensors, OS versions, wireless data rates, and system configurations. Each can lead to different and unpredictable behavior, but testing applications across all environments is an impossible task.

Android has therefore been constructed to ensure as uniform an experience across platforms as possible. By abstracting the hardware differences, Android OS tries to insulate applications from device-specific modifications while providing the flexibility to tune aspects as needed. Future-proofing of applications to the introduction of new hardware platforms and OS updates is also a consideration. This mostly works as long as the developer is well aware of this systematic approach. The generic Application Programming Interfaces (API) that Android offers and how to ensure device and OS compatibility are main threads discussed throughout this book.

Still, as with any embedded platform, extensive testing of applications is required. Google provides assistance to third-party developers in many forms as Android Development Tool (ADT) plugins for Eclipse (also as standalone tools) including real-time logging capabilities, a realistic emulator that runs native ARM code, and in-field error reports from users to developers of Android Market applications.

The Dichotomy of Android

Android has some interesting dichotomies. Knowing about them upfront is useful not only in understanding what Android is, but what it is not.

Android is an embedded OS that relies on the Linux kernel for core system services, but it is not embedded Linux. For example, standard Linux utilities such as X-windows and GNU C libraries are not supported. Writing applications for Android utilizes the Java framework, but it is not Java. Standard Java libraries such as Swing are not supported. Other libraries such as Timer are not preferred; they have been replaced by Android's own libraries, which are optimized for usage in a resource-constrained, embedded environment.

The Android OS is open source, which means developers can view and use any of the system source code, including the radio stack. This source code is one of the first resources for seeing examples of Android code in action, and it helps clarify the usage when documentation is lacking. This also means developers can utilize the system in the same way as any core application and can swap out system components for their own components. However, Android devices do contain some proprietary software that is inaccessible to developers (such as Global Positioning System (GPS) navigation).

A final dichotomy of Android OS is that Google is also backing Chrome OS. Android OS is built for embedded platforms, and Chrome OS is built for cloud-based platforms. However, which is the best choice for embedded devices that live in the cloud? Netbooks, which fill the gap between smart phones and laptop computers, could presumably go either way (and they have). Android has started to utilize the cloud more. Does that mean Chrome OS's days are numbered? Google also backs a web-based market, so Chrome OS enjoys the same developer leverage that Android currently has. This points to a convergence that might have been in the cards all along.

Devices Running Android

There are more than 40 Android phones in the market from more than ten manufacturers. Other hardware also runs Android, such as tablets and televisions. Software can access information on the target device using the `android.os.Build` class, for example:

```
if(android.os.Build.MODEL.equals("Nexus+One")) { ... }
```

Android-supported hardware shares some common features due to the nature of the operating system. The Android OS is organized into the following images:

- Bootloader—Initiates loading of the boot image during startup
- Boot image—Kernel and RAMdisk
- System image—Android operating system platform and apps
- Data image—User data saved across power cycles
- Recovery image—Files used for rebuilding or updating the system
- Radio image—Files of the radio stack

These images are stored on nonvolatile flash memory, so they are protected when the device powers down. The flash memory is used like read-only memory (hence, some call it ROM), but can it be rewritten as necessary (for example, with over-the-air Android operating system updates).

On startup, the microprocessor executes the bootloader to load the kernel and RAMdisk to RAM for quick access. The microprocessor then executes instructions and pages portions of the system and data images into RAM as needed. The radio image resides on the baseband processor, which connects to the radio hardware.

A comparison of some of the early and more recent smart phone models is shown in Table 1.1. It shows that the processing hardware architecture is similar across devices: a microprocessor unit (MPU), synchronous dynamic random access memory (SDRAM or RAM for short), and flash memory (called ROM for short). The screen size is given in pixels, but the dots per inch (dpi) vary depending on the physical screen size. For example, the HTC Magic has a 3.2-inch diagonal screen with 320x480 pixels. This equates to 180 pixels per inch, but is classified as a medium pixel density device by Android (which averages as 160 dpi). All smartphones also offer a CMOS image sensor camera, Bluetooth (BT), and Wi-Fi (802.11), although there are variations.

Table 1.1 **Comparison of Some Representative Android Smartphones. Data from http://en.wikipedia.org/wiki/List_of_Android_devices and http://pdadb.net/.**

Model	MPU	RAM/ ROM	Screen	Other Features
HTC Dream / G1 (October 2008)	528-MHz QCOM MSM7201A	192MB/ 256MB	TFT LCD 320x480 mdpi	GSM/UMTS slide out keyboard, trackball, AGPS BT2.0, 802.11b/g, 3.1-MP camera

Table 1.1 **Continued**

Model	MPU	RAM/ ROM	Screen	Other Features
Samsung Moment (November 2009)	800-MHz ARM1176 JZF-S	288MB/ 512MB	AMOLED 320x480 mdpi	CDMA/1xEV-DO slide out keyboard (backlit), DPAD BT2.0, 802.11b/g, 3.1-MP camera AGPS
Motorola Milestone / Droid (November 2009)	550-MHz TI OMAP3430	256MB/ 512MB	TFT LCD 480x854 hdpi	GSM/UMTS or CDMA/1xEV-DO slide out keyboard, DPAD BT2.1, 802.11b/g, 5-MP camera AGPS
Nexus One / HTC Passion (January 2010)	1-GHz QCOM Snapdragon	512MB/ 512MB	AMOLED 480x800 hdpi	GSM/UMTS Trackball, dual microphones BT2.0, 802.11a/b/g/n, 5-MP camera AGPS, geotagging
HTC Droid Incredible (April 2010)	1-GHz QCOM Snapdragon	512MB/ 512MB	AMOLED 480x800 hdpi	CDMA/1xEV-DO BT2.1, 802.11a/b/g/n, 8-MP camera AGPS, geotagging
HTC EVO 4G (June 2010)	1-GHz QCOM Snapdragon	512MB/ 1GB	TFT LCD 480x800 hdpi	CDMA/1xEV-DO/802.16e-2005 BT2.1, 802.11b/g, 8-MP camera 1.3MP front-facing camera, AGPS

Table 1.1 **Continued**

Model	MPU	RAM/ ROM	Screen	Other Features
Motorola Droid X (July 2010)	1-GHz TI OMAP3630	512MB/ 8GB	TFT LCD 480x854 hdpi	CDMA/1xEV-DO, FM radio BT2.1, 802.11b/g/n, 8-MP camera AGPS, geotagging
Sony-Ericsson Xperia X10a (June 2010)	1-GHz QCOM Snapdragon	256MB/ 1GB	TFT LCD 480x854 hdpi	GSM/UMTS, FM radio BT2.1, 802.11b/g, 8-MP camera AGPS, geotagging
Samsung Galaxy S Pro (August 2010)	1-GHz Samsung Hummingbird	512MB/ 2GB	AMOLED 480x800 hdpi	CDMA/1xEV-DO, 802.16, FM radio slide out keyboard BT3.0, 802.11b/g/n, 5-MP camera 0.3MP front-facing camera, AGPS
Acer Stream / Liquid (September 2010)	1-GHz QCOM Snapdragon	512MB/ 512MB	AMOLED 480x800 hdpi	GSM/UMTS, FM radio BT2.1, 802.11b/g/n, 5-MP camera AGPS, geotagging

Other than improved capacity and performance on newer models, another main differentiator is additional features. Some devices offer 4G, some have FM radio, some have slide-out keyboards, and some have a front-facing camera. Knowing the differentiators helps a developer create great applications. In addition to the built-in hardware, every Android device comes with a secure digital (SD) card slot. An SD card provides additional storage space for multimedia and extra application data. However, until Android 2.2, the apps themselves could be stored only on the internal ROM.

HTC Models

HTC is a Taiwanese company founded in 1997. The first commercially available hardware running Android was the HTC Dream (also known as the G1 with G standing for Google). It was released in October 2008. Since then, HTC has put out over ten phones running Android, including Google's Nexus One.

The Nexus One was one of the first Android devices to use a 1-GHz microprocessor, the Snapdragon platform from Qualcomm. The Snapdragon includes Qualcomm's own core as opposed to an ARM core, and it contains circuitry to decode high-definition video at 720p. Most smartphones that have followed also utilize a 1-GHz microprocessor. Other distinctions of the Nexus One are the use of two microphones to cancel background noise during phone conversations and a backlit trackball that lights up different colors based on the notification.

HTC also released the Droid Incredible in April 2010. As seen in Table 1.1, it is similar to the Nexus One but has a CDMA instead of a GSM radio hardware and a higher pixel density camera. The HTC EVO 4G released in June 2010 produced quite a sensation as the first commercially available phone that supports WiMAX (802.16e-2005).

Motorola Models

Motorola built the first cell phone in the 1980s and has had diverse success in the cell phone market since. More recently, the wireless division was wavering for a direction until it focused efforts on Android. The release of the Motorola Droid for CDMA (also known as the Milestone for the GSM worldwide version) in November 2009 is indeed considered by many as a major milestone for Android. The Droid's impact is apparent in that a significant fraction of Android phones accessing the Android Market are Droids.

In addition, Motorola has put out close to ten additional phone brands running Android. The Motorola Droid X has capabilities similar to the HTC Droid Incredible, including HD video capture.

Samsung Models

Samsung has been a strong force in the mobile market and is starting to come into its own with Android devices. The Samsung Moment was introduced in November 2009, but does not have hardware capability for multitouch. It will not be upgraded beyond Android 2.1. A custom version, including a Mobile TV antenna, is available in select markets for receiving Mobile ATSC signals.

The Samsung Galaxy S is Samsung's answer to the iPhone. It is well known that Samsung processors are used in the iPhone 3G and 3GS. With the Galaxy S, Samsung developed a 1-GHz Hummingbird processor with an ARM Cortex-8 core. It is also one of the first phones to offer Bluetooth 3.0 compatibility.

Tablets

With Apple's introduction of the iPad, Android manufacturers were expected to introduce tablet computers of their own. A tablet computer is loosely defined as having a screen of 4.8 inches or larger and Wi-Fi connectivity. Because many have 3G wireless service, they tend to be more like smartphones with large screens.

Archos was one of the first to market an Android tablet in late 2009. It has a diagonal screen size of 4.8 inches and is called the Archos 5. Archos has since introduced a 7-inch model called the Archos 7. These models come with an actual hard drive for more data storage. Dell has also introduced a 5-inch tablet called the Streak with plans for both a 7-inch and a 10-inch screen size model. Samsung offers the Galaxy Tab with a 7-inch screen. One downside is the inability for many of these tablets to access the Android Market, although that should soon change. A comparison of some tablet computer models is shown in Table 1.2.

Table 1.2 **Comparison of Representative Android Tablet Computers**

Model	MPU	RAM/ disk	Screen	Other Features
Archos 5 (September 2009)	800-MHz TI OMAP 3440	256MB/ 8GB	TFT LCD 4.8 inches 800x480	BT2.0, 802.11b/g/n, FM radio
Archos 7 (June 2010)	600-MHz Rockchip RK2808	128MB/ 8GB	TFT LCD 7 inches 800x480	802.11b/g
Dell Streak (June 2010)	1-GHz QCOM Snapdragon	256MB/ 512MB	TFT LCD 5 inches 800x480	GSM/UMTS, BT2.1, 802.11b/g, 5-MP camera, 0.3-MP front-facing camera AGPS, geotagging
Samsung Galaxy Tablet GT-P1000 (September 2010)	1-GHz Samsung Hummingbird	512MB/ 16GB	TFT LCD 7 inches 1024x600	GSM/UMTS BT3.0, 802.11b/g/n, 3.1-MP camera

Other Devices

Given Android is a generic embedded platform, it is expected to be utilized in many other industries beyond smartphones and tablet computers. The first Android-based automobile is the Roewe 350, which Shanghai Automotive Industry Corporation manufactures. Android is mainly used for GPS navigation but can also support web browsing.