

Begin the journey towards building  
your own Android 4 apps

# Beginning Android 4

Grant Allen

Apress®



# Beginning Android 4



**Grant Allen**

**Apress®**

## **Beginning Android 4**

Copyright © 2012 by Grant Allen

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-4302-3984-0

ISBN-13 (electronic): 978-1-4302-3985-7

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The images of the Android Robot (01 / Android Robot) are reproduced from work created and shared by Google and used according to terms described in the Creative Commons 3.0 Attribution License. Android and all Android and Google-based marks are trademarks or registered trademarks of Google, Inc., in the U.S. and other countries. Apress Media, L.L.C. is not affiliated with Google, Inc., and this book was written without endorsement from Google, Inc.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

President and Publisher: Paul Manning

Lead Editor: Steve Anglin

Development Editor: Jonathan Gennick

Technical Reviewers: Nikhil Gopal and Michael Thomas

Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Morgan Ertel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh

Coordinating Editor: Jessica Belanger

Copy Editor: William McManus

Compositor: MacPS, LLC

Indexer: John Collin

Artist: April Milne

Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media, LLC., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com).

For information on translations, please e-mail [rights@apress.com](mailto:rights@apress.com), or visit [www.apress.com](http://www.apress.com).

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at [www.apress.com/bulk-sales](http://www.apress.com/bulk-sales).

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

Any source code or other supplementary materials referenced by the author in this text is available to readers at [www.apress.com](http://www.apress.com). For detailed information about how to locate your book’s source code, go to <http://www.apress.com/source-code/>.

---

# Contents at a Glance

<b>Contents</b> .....	<b>V</b>
<b>About the Authors</b> .....	<b>xvii</b>
<b>About the Technical Reviewer</b> .....	<b>xviii</b>
<b>Acknowledgments</b> .....	<b>xix</b>
<b>Preface</b> .....	<b>xx</b>
<b>Part I: Core Concept</b> .....	<b>1</b>
■ <b>Chapter 1: The Big Picture</b> .....	<b>3</b>
■ <b>Chapter 2: How to Get Started</b> .....	<b>7</b>
■ <b>Chapter 3: Your First Android Project</b> .....	<b>23</b>
■ <b>Chapter 4: Examining Your First Project</b> .....	<b>31</b>
■ <b>Chapter 5: A Bit About Eclipse</b> .....	<b>37</b>
■ <b>Chapter 6: Enhancing Your First Project</b> .....	<b>47</b>
<b>Part II: Activities</b> .....	<b>51</b>
■ <b>Chapter 7: Rewriting Your First Project</b> .....	<b>53</b>
■ <b>Chapter 8: Using XML-Based Layouts</b> .....	<b>57</b>
■ <b>Chapter 9: Employing Basic Widgets</b> .....	<b>63</b>
■ <b>Chapter 10: Working with Containers</b> .....	<b>79</b>
■ <b>Chapter 11: The Input Method Framework</b> .....	<b>103</b>
■ <b>Chapter 12: Using Selection Widgets</b> .....	<b>113</b>
■ <b>Chapter 13: Getting Fancy with Lists</b> .....	<b>129</b>
■ <b>Chapter 14: Still More Widgets and Containers</b> .....	<b>145</b>
■ <b>Chapter 15: Embedding the WebKit Browser</b> .....	<b>169</b>
■ <b>Chapter 16: Applying Menus</b> .....	<b>177</b>
■ <b>Chapter 17: Showing Pop-Up Messages</b> .....	<b>189</b>
■ <b>Chapter 18: Handling Activity Lifecycle Events</b> .....	<b>193</b>
■ <b>Chapter 19: Handling Rotation</b> .....	<b>197</b>
■ <b>Chapter 20: Dealing with Threads</b> .....	<b>213</b>

■ Chapter 21: Creating Intent Filters .....	231
■ Chapter 22: Launching Activities and Subactivities.....	237
■ Chapter 23: Working with Resources .....	245
■ Chapter 24: Defining and Using Styles .....	263
<b>Part III: Honeycomb and Tablets.....</b>	<b>269</b>
■ Chapter 25: Handling Multiple Screen Sizes .....	271
■ Chapter 26: Focusing on Tablets and Larger UIs.....	293
■ Chapter 27: Using the Action Bar .....	299
■ Chapter 28: Fragments .....	307
■ Chapter 29: Handling Platform Changes .....	323
<b>Part IV: Data Stores, Network Services, and APIs .....</b>	<b>333</b>
■ Chapter 30: Accessing Files .....	335
■ Chapter 31: Using Preferences .....	349
■ Chapter 32: Managing and Accessing Local Databases.....	367
■ Chapter 33: Leveraging Java Libraries .....	381
■ Chapter 34: Communicating via the Internet .....	389
<b>Part V: Services .....</b>	<b>407</b>
■ Chapter 35: Services: The Theory.....	409
■ Chapter 36: Basic Service Patterns.....	417
■ Chapter 37: Alerting Users via Notifications .....	437
<b>Part VI: Other Android Capabilities.....</b>	<b>449</b>
■ Chapter 38: Requesting and Requiring Permissions .....	451
■ Chapter 39: Accessing Location-Based Services .....	457
■ Chapter 40: Mapping with MapView and MapActivity .....	463
■ Chapter 41: Handling Telephone Calls.....	477
■ Chapter 42: Fonts .....	481
■ Chapter 43: More Development Tools .....	487
<b>Part VII: Alternative Application Environments .....</b>	<b>505</b>
■ Chapter 44: The Role of Alternative Environments.....	507
■ Chapter 45: HTML5 .....	511
■ Chapter 46: PhoneGap .....	525
■ Chapter 47: Other Alternative Environments.....	543
<b>Part VIII: The Ever-Evolving Android .....</b>	<b>549</b>
■ Chapter 48: Dealing with Devices.....	551
■ Chapter 49: Where Do We Go from Here? .....	557
<b>Index.....</b>	<b>561</b>

---

# Contents

<b>Contents at a Glance</b> .....	<b>iii</b>
<b>About the Author</b> .....	<b>xvii</b>
<b>About the Technical Reviewers</b> .....	<b>xviii</b>
<b>Acknowledgments</b> .....	<b>xix</b>
<b>Preface</b> .....	<b>xx</b>
<b>Part I: Core Concept</b> .....	<b>1</b>
■ <b>Chapter 1: The Big Picture</b> .....	<b>3</b>
Benefits and Drawbacks of Smartphone Programming.....	3
What Androids Are Made Of.....	4
Stuff at Your Disposal .....	5
The Big Picture...of This Book .....	6
■ <b>Chapter 2: How to Get Started</b> .....	<b>7</b>
Step 1: Set Up Java.....	7
Install the JDK.....	7
Learn Java .....	8
Step 2: Install the Android SDK.....	9
Install the Base Tools.....	9
Install the SDKs and Add-ons .....	9
Step 3: Install the ADT for Eclipse.....	12
Step 4: Install Apache Ant.....	14
Step 5: Set Up the Emulator.....	15
Step 6: Set Up the Device .....	21
Windows .....	22
Mac OS X and Linux.....	22
■ <b>Chapter 3: Your First Android Project</b> .....	<b>23</b>
Step 1: Create the New Project.....	23
Eclipse .....	23
Command Line .....	26
Step 2: Build, Install, and Run the Application in Your Emulator or Device .....	27
Eclipse .....	27
Command Line .....	28

■ <b>Chapter 4: Examining Your First Project .....</b>	<b>31</b>
Project Structure .....	31
Root Contents .....	31
The Sweat Off Your Brow .....	32
And Now, the Rest of the Story .....	32
What You Get Out of It .....	33
Inside Your Manifest .....	33
In the Beginning, There Was the Root, and It Was Good .....	34
An Application for Your Application .....	35
■ <b>Chapter 5: A Bit About Eclipse.....</b>	<b>37</b>
What the ADT Gives You.....	37
Coping with Eclipse .....	38
How to Import a Non-Eclipse Project.....	38
How to Get to DDMS .....	42
How to Create an Emulator .....	44
How to Run a Project .....	44
How Not to Run Your Project .....	45
Alternative IDEs.....	45
IDEs and This Book .....	46
■ <b>Chapter 6: Enhancing Your First Project .....</b>	<b>47</b>
Supporting Multiple Screen Sizes.....	47
Specifying Versions .....	48
<b>Part II: Activities .....</b>	<b>51</b>
■ <b>Chapter 7: Rewriting Your First Project .....</b>	<b>53</b>
The Activity .....	53
Dissecting the Activity .....	54
Building and Running the Activity.....	55
■ <b>Chapter 8: Using XML-Based Layouts .....</b>	<b>57</b>
What Is an XML-Based Layout? .....	57
Why Use XML-Based Layouts? .....	57
OK, So What Does It Look Like?.....	58
What's with the @ Signs?.....	59
And How Do We Attach These to the Java? .....	59
The Rest of the Story .....	60
■ <b>Chapter 9: Employing Basic Widgets.....</b>	<b>63</b>
Assigning Labels .....	63
Button, Button, Who's Got the Button? .....	64
Fleeting Images .....	65
Fields of Green...or Other Colors.....	66
Just Another Box to Check.....	68
Throw the Switch, Igor.....	71
Turn Up the Radio .....	73
It's Quite a View .....	75
Padding.....	75
Other Useful Properties.....	76
Useful Methods .....	76

Colors.....	76
<b>Chapter 10: Working with Containers .....</b>	<b>79</b>
Thinking Linearly .....	79
LinearLayout Concepts and Properties .....	80
LinearLayout Example.....	82
The Box Model .....	86
All Things Are Relative .....	88
RelativeLayout Concepts and Properties .....	88
RelativeLayout Example.....	90
Overlap.....	92
Tabula Rasa .....	93
TableLayout Concepts and Properties .....	94
TableLayout Example.....	95
Scrollwork.....	96
Take Them to the Grid.....	99
<b>Chapter 11: The Input Method Framework.....</b>	<b>103</b>
Keyboards, Hard and Soft .....	103
Tailored to Your Needs.....	104
Tell Android Where It Can Go .....	108
Fitting In.....	110
Jane, Stop This Crazy Thing! .....	111
<b>Chapter 12: Using Selection Widgets .....</b>	<b>113</b>
Adapting to the Circumstances.....	113
Lists of Naughty and Nice .....	114
Selection Modes .....	116
Spin Control .....	118
Grid Your Lions (or Something Like That...) .....	121
Fields: Now with 35% Less Typing! .....	125
Galleries, Give or Take the Art .....	128
<b>Chapter 13: Getting Fancy with Lists .....</b>	<b>129</b>
Getting to First Base .....	129
A Dynamic Presentation.....	131
Inflating Rows Ourselves .....	133
A Sidebar About Inflation .....	133
And Now, Back to Our Story .....	135
Better. Stronger. Faster. ....	135
Using convertView .....	135
Using the Holder Pattern.....	137
Interactive Rows .....	139
<b>Chapter 14: Still More Widgets and Containers.....</b>	<b>145</b>
Pick and Choose .....	145
Time Keeps Flowing Like a River.....	150
Seeking Resolution .....	152
Putting It on My Tab.....	152
The Pieces .....	153
Wiring It Together .....	154



Adding Them Up.....	157
Flipping Them Off.....	160
Getting in Somebody’s Drawer .....	165
Other Good Stuff .....	167
<b>■ Chapter 15: Embedding the WebKit Browser .....</b>	<b>169</b>
A Browser, Writ Small.....	169
Loading It Up.....	171
Navigating the Waters.....	172
Entertaining the Client .....	173
Settings, Preferences, and Options (Oh, My!) .....	175
<b>■ Chapter 16: Applying Menus .....</b>	<b>177</b>
Flavors of Menu .....	177
Menus of Options.....	178
Menus in Context.....	179
Taking a Peek .....	180
Yet More Inflation.....	185
Menu XML Structure .....	185
Menu Options and XML .....	186
Inflating the Menu.....	187
When Giant Menus Walk the Earth .....	188
<b>■ Chapter 17: Showing Pop-Up Messages.....</b>	<b>189</b>
Raising Toasts.....	189
Alert! Alert!.....	190
Checking Them Out.....	191
<b>■ Chapter 18: Handling Activity Lifecycle Events .....</b>	<b>193</b>
Schrödinger’s Activity.....	193
Life, Death, and Your Activity.....	194
onCreate() and onDestroy() .....	194
onStart(), onRestart(), and onStop() .....	195
onPause() and onResume() .....	195
The Grace of State .....	195
<b>■ Chapter 19: Handling Rotation .....</b>	<b>197</b>
A Philosophy of Destruction.....	197
It’s All the Same, Just Different .....	198
Picking and Viewing a Contact .....	199
Saving Your State .....	200
Now with More Savings!.....	203
DIY Rotation .....	205
Forcing the Issue .....	208
Making Sense of It All .....	210
<b>■ Chapter 20: Dealing with Threads .....</b>	<b>213</b>
The Main Application Thread .....	213
Making Progress with ProgressBars.....	214
Getting Through the Handlers .....	214
Messages.....	215
Runnables .....	218

Where Oh Where Has My UI Thread Gone? .....	218
Asynching Feeling.....	218
The Theory .....	218
AsyncTask, Generics, and Varargs.....	219
The Stages of AsyncTask.....	220
A Sample Task .....	220
Threads and Rotation.....	224
Manual Activity Association .....	225
Flow of Events .....	227
Why This Works .....	228
And Now, the Caveats.....	228
<b>■ Chapter 21: Creating Intent Filters .....</b>	<b>231</b>
What's Your Intent?.....	231
Pieces of Intents .....	232
Intent Routing .....	232
Stating Your Intent(ions) .....	233
Narrow Receivers .....	234
Intents for Every Occasion .....	235
The Pause Caveat .....	235
<b>■ Chapter 22: Launching Activities and Subactivities.....</b>	<b>237</b>
Peers and Subs .....	237
Start 'Em Up.....	238
Make an Intent.....	238
Make the Call .....	238
Tabbed Browsing, Sort Of.....	242
<b>■ Chapter 23: Working with Resources .....</b>	<b>245</b>
The Resource Lineup .....	245
String Theory .....	246
Plain Strings.....	246
String Formats .....	246
Styled Text .....	247
Styled Text and Formats .....	247
Got the Picture? .....	251
XML: The Resource Way .....	252
Miscellaneous Values .....	254
Dimensions .....	255
Colors.....	255
Arrays.....	256
Different Strokes for Different Folks .....	257
RTL Languages: Going Both Ways .....	260
<b>■ Chapter 24: Defining and Using Styles .....</b>	<b>263</b>
Styles: DIY DRY .....	263
Elements of Style .....	265
Where to Apply a Style.....	265
The Available Attributes .....	266

Inheriting a Style .....	266
The Possible Values .....	267
Themes: A Style by Any Other Name .....	268
<b>Part III: Honeycomb and Tablets.....</b>	<b>269</b>
<b>■ Chapter 25: Handling Multiple Screen Sizes .....</b>	<b>271</b>
Taking the Default.....	271
Whole in One.....	273
Think About Rules, Not Positions .....	273
Consider Physical Dimensions .....	274
Avoid “Real” Pixels .....	274
Choose Scalable Drawables .....	274
Tailor-Made, Just for You (and You, and You, and...).....	275
Adding the <supports-screens> Element.....	275
Resources and Resource Sets .....	276
Finding Your Size .....	277
Ain’t Nothing Like the Real Thing.....	278
Density Differs .....	278
Adjusting the Density .....	278
Ruthlessly Exploiting the Situation .....	279
Replace Menus with Buttons .....	280
Replace Tabs with a Simple Activity .....	280
Consolidate Multiple Activities.....	280
Example: EU4You .....	281
The First Cut.....	281
Fixing the Fonts .....	287
Fixing the Icons.....	289
Using the Space .....	289
What If It Is Not a Browser? .....	292
<b>■ Chapter 26: Focusing on Tablets and Larger UIs.....</b>	<b>293</b>
Why the Drive to Tablets? .....	293
What the User Sees.....	294
Dealing with the Rest of the Devices .....	297
<b>■ Chapter 27: Using the Action Bar .....</b>	<b>299</b>
Enabling the Action Bar .....	299
Promoting Menu Items to the Action Bar.....	300
Responding to the Logo .....	301
Adding Custom Views to the Action Bar .....	301
Defining the Layout.....	302
Putting the Layout in the Menu.....	303
Getting Control of User Input.....	304
Don’t Forget the Phones! .....	305
<b>■ Chapter 28: Fragments.....</b>	<b>307</b>
Introducing Fragments.....	307
The Problem Addressed by Fragments .....	307
The Fragments Solution .....	308
The Android Compatibility Library.....	309
Creating Fragment Classes.....	310

General Fragments .....	310
ListFragment.....	311
Other Fragment Base Classes.....	316
Fragments, Layouts, Activities, and Multiple Screen Sizes .....	316
EU4You.....	317
DetailsActivity .....	321
Fragments and Configuration Changes.....	322
Designing for Fragments .....	322
■ <b>Chapter 29: Handling Platform Changes .....</b>	<b>323</b>
Things That Make You Go Boom .....	323
View Hierarchy.....	323
Changing Resources .....	324
Handling API Changes.....	324
Minimum, Maximum, Target, and Build Versions .....	325
Detecting the Version.....	327
Wrapping the API .....	327
Patterns for Ice Cream Sandwich and Honeycomb .....	329
The Action Bar .....	329
Writing Tablet-Only Apps .....	331
<b>Part IV: Data Stores, Network Services, and APIs .....</b>	<b>333</b>
■ <b>Chapter 30: Accessing Files .....</b>	<b>335</b>
You and the Horse You Rode in On .....	335
Readin' 'n Writin' .....	338
External Storage: Giant Economy-Size Space .....	342
Where to Write .....	342
When to Write .....	343
StrictMode: Avoiding Janky Code .....	343
Setting Up StrictMode.....	344
Seeing StrictMode in Action .....	344
Development Only, Please! .....	345
Conditionally Being Strict.....	345
Linux File Systems: You Sync, You Win .....	347
■ <b>Chapter 31: Using Preferences .....</b>	<b>349</b>
Getting What You Want .....	349
Stating Your Preference.....	350
Introducing PreferenceFragment and PreferenceActivity .....	350
Preferences via Fragments .....	351
Preferences the New and Improved Way.....	351
Preference Headers .....	351
PreferenceFragment and StockPreferenceFragment .....	352
Avoiding Nested PreferenceScreen Elements .....	354
Intents for Headers or Preferences .....	354
Adding Backward Compatibility.....	354
The Older Model of Preference Handling .....	355
Letting Users Have Their Say.....	356
Adding a Wee Bit o' Structure.....	360
The Kind of Pop-Ups You Like.....	362

■ <b>Chapter 32: Managing and Accessing Local Databases.....</b>	<b>367</b>
A Quick SQLite Primer.....	369
Start at the Beginning.....	369
Setting the Table.....	372
Makin' Data.....	373
What Goes Around, Comes Around.....	374
Raw Queries.....	374
Regular Queries.....	374
Using Cursors.....	375
Custom CursorAdapters.....	376
Making Your Own Cursors.....	377
SQLite and Android Versions.....	377
Flash: Sounds Faster Than It Is.....	377
Ship Ahoy!.....	378
Data, Data, Everywhere.....	378
■ <b>Chapter 33: Leveraging Java Libraries .....</b>	<b>381</b>
Ants and JARs.....	381
The Outer Limits.....	382
Following the Script.....	383
Reviewing the Script.....	386
■ <b>Chapter 4: Communicating via the Internet .....</b>	<b>389</b>
REST and Relaxation.....	389
HTTP Operations via Apache HttpClient.....	390
Parsing Responses.....	391
Stuff to Consider.....	394
AndroidHttpClient.....	394
Leveraging Internet-Aware Android Components.....	394
Downloading Files.....	395
Continuing Our Escape from Janky Code.....	405
<b>Part V: Services .....</b>	<b>407</b>
■ <b>Chapter 35: Services: The Theory.....</b>	<b>409</b>
Why Services?.....	409
Setting Up a Service.....	410
Service Class.....	410
Lifecycle Methods.....	410
Manifest Entry.....	411
Communicating to Services.....	411
Sending Commands with startService().....	412
Binding with bindService().....	413
Communicating from Services.....	414
Callback/Listener Objects.....	414
Broadcast Intents.....	414
Pending Results.....	415
Messenger.....	415
Notifications.....	416

■ <b>Chapter 36: Basic Service Patterns</b> .....	<b>417</b>
The Downloader .....	417
The Design .....	417
The Service Implementation .....	418
Using the Service .....	420
The Music Player .....	421
The Design .....	421
The Service Implementation .....	422
Using the Service .....	423
The Web Service Interface .....	424
The Design .....	425
The Rotation Challenge .....	425
The Service Implementation .....	425
Using the Service .....	430
■ <b>Chapter 37: Alerting Users via Notifications</b> .....	<b>437</b>
Notification Configuration .....	437
Hardware Notifications .....	438
Icons .....	438
Notifications in Action .....	440
Staying in the Foreground .....	444
FakePlayer, Redux .....	445
Notifications in Ice Cream Sandwich and Honeycomb .....	446
<b>Part VI: Other Android Capabilities</b> .....	<b>449</b>
■ <b>Chapter 38: Requesting and Requiring Permissions</b> .....	<b>451</b>
Mother, May I? .....	451
Halt! Who Goes There? .....	452
Enforcing Permissions via the Manifest .....	453
Enforcing Permissions Elsewhere .....	454
May I See Your Documents? .....	454
New Permissions in Old Applications .....	455
Permissions: Up Front or Not at All .....	455
■ <b>Chapter 39: Accessing Location-Based Services</b> .....	<b>457</b>
Location Providers: They Know Where You're Hiding .....	457
Finding Yourself .....	458
On the Move .....	459
Are We There Yet? Are We There Yet? .....	460
Testing... Testing... .....	461
■ <b>Chapter 40: Mapping with MapView and MapActivity</b> .....	<b>463</b>
Terms, Not of Endearment .....	463
Piling On .....	464
The Key to It All .....	464
The Bare Bones .....	465
Optional Maps .....	466
Exercising Your Control .....	467
Zoom .....	467
Center .....	467

Layers Upon Layers.....	468
Overlay Classes.....	468
Drawing the ItemizedOverlay.....	468
Handling Screen Taps.....	470
My, Myself, and MyLocationOverlay.....	470
Rugged Terrain.....	472
Maps and Fragments.....	473
Limit Yourself to the Latest Android Versions.....	473
Use onCreateView() and onActivityCreated().....	474
Host the Fragment in a MapActivity.....	475
A Custom Alternative for Maps and Fragments.....	476
<b>■ Chapter 41: Handling Telephone Calls.....</b>	<b>477</b>
Report to the Manager.....	477
You Make the Call!.....	478
No, Really, You Make the Call!.....	480
<b>■ Chapter 42: Fonts.....</b>	<b>481</b>
Love the One You're With.....	481
Additional Fonts.....	483
Here a Glyph, There a Glyph.....	484
<b>■ Chapter 43: More Development Tools.....</b>	<b>487</b>
Hierarchy Viewer: How Deep Is Your Code?.....	487
DDMS: Under Android's Hood.....	491
Logging.....	493
File Push and Pull.....	494
Screenshots.....	495
Location Updates.....	496
Placing Calls and Messages.....	496
Memory Management.....	499
adb: Like DDMS, with More Typing.....	502
Getting Graphical.....	502
<b>Part VII: Alternative Application Environments.....</b>	<b>505</b>
<b>■ Chapter 44: The Role of Alternative Environments.....</b>	<b>507</b>
In the Beginning, There Was Java.....	507
...And It Was OK.....	508
Bucking the Trend.....	508
Support, Structure.....	509
Caveat Developer.....	509
<b>■ Chapter 45: HTML5.....</b>	<b>511</b>
Offline Applications.....	511
What Does It Mean?.....	511
How Do You Use It?.....	512
Web Storage.....	517
What Does It Mean?.....	517
How Do You Use It?.....	518
Web SQL Database.....	519
Going to Production.....	519

Testing .....	520
Signing and Distribution .....	520
Updates .....	520
Issues You May Encounter .....	520
Android Device Versions .....	520
Screen Sizes and Densities .....	521
Limited Platform Integration .....	521
Performance and Battery .....	522
Look and Feel .....	522
Distribution .....	522
Browser Changes Post Ice Cream Sandwich .....	523
HTML5 and Alternative Android Browsers .....	523
HTML5: The Baseline .....	523
<b>■ Chapter 46: PhoneGap .....</b>	<b>525</b>
What Is PhoneGap? .....	525
What Do You Write In? .....	525
What Features Do You Get? .....	526
What Do Apps Look Like? .....	526
How Does Distribution Work? .....	527
What About Other Platforms? .....	527
Using PhoneGap .....	528
Installation .....	528
Creating and Installing Your Project .....	528
PhoneGap Build .....	529
PhoneGap and the Checklist Sample .....	534
Sticking to the Standards .....	534
Adding PhoneGap APIs .....	536
Issues You May Encounter .....	539
Security .....	539
Screen Sizes and Densities .....	540
Look and Feel .....	541
For More Information .....	541
<b>■ Chapter 47: Other Alternative Environments .....</b>	<b>543</b>
Rhodes .....	543
Flash, Flex, and AIR .....	544
JRuby and Ruboto .....	544
Mono for Android .....	545
App Inventor .....	545
Titanium Mobile .....	548
Other JVM Compiled Languages .....	548
<b>Part VIII: The Ever-Evolving Android .....</b>	<b>549</b>
<b>■ Chapter 48: Dealing with Devices .....</b>	<b>551</b>
This App Contains Explicit Instructions .....	551
Explicit Feature Requests .....	551
Implied Feature Requests .....	553
A Guaranteed Market .....	553
Other Stuff That Varies .....	554



Bugs, Bugs, Bugs.....	554
Device Testing .....	555
<b>■ Chapter 49: Where Do We Go from Here? .....</b>	<b>557</b>
Questions, Sometimes with Answers .....	557
Heading to the Source .....	558
Getting Your News Fix.....	559
<b>Index .....</b>	<b>561</b>

---

# About the Author



**Grant Allen** has worked in the IT field for over 20 years as a CTO, enterprise architect, and database architect. Grant's roles have covered private enterprise, academia, and the government sector around the world, specializing in global-scale systems design, development, and performance. He is a frequent speaker at industry and academic conferences, on topics ranging from data mining to compliance, and technologies such as databases (DB2, Oracle, SQL Server, and MySQL), content management, collaboration, disruptive innovation, and mobile ecosystems like Android.

His first Android application was a task list to remind him to finish all his other unfinished Android projects.

Grant works for Google, and in his spare time is completing a PhD on building innovative high-technology environments.

Grant is the author of *Beginning DB2: From Novice to Professional* (Apress, 2008), and lead author of *Oracle SQL Recipes: A Problem-Solution Approach* (Apress, 2010) and *The Definitive Guide to SQLite*, 2nd Edition (Apress, 2010).

---

# About the Technical Reviewers

**Michael Thomas** has worked in software development for over 20 years as an individual contributor, team lead, program manager, and Vice President of Engineering. Michael has over 10 years experience working with mobile devices. His current focus is in the medical sector using mobile devices to accelerate information transfer between patients and health care providers.

**Nikhil Gopal** is a director of software development at Intelligene, LLC. He works on machine learning and genomics/bioinformatics applications. In the past, he has worked in the biotechnology and software industries, primarily in the role of a programmer scientist. He has contributed to a number of open source projects pertaining to bioinformatics, genomics, and unix-based operating systems. He graduated from UC Davis with a B.S. in Biotechnology/Bioinformatics. He enjoys playing the guitar, riding motorcycles, and a number of outdoor activities. He currently lives in the San Francisco bay area.

---

# Acknowledgments

I would like to thank the Android team, not only for putting out a good product, but for invaluable assistance on the Android Google Groups.

Some of the icons used in the sample code were provided by the Nuvola icon set.

---

# Preface

## Welcome to the Book!

Thanks for your interest in developing applications for Android! Increasingly, people will access Internet-based services using so-called “nontraditional” means, such as mobile devices. The more we do in that space now, the more that people will help invest in that space to make it easier to build more powerful mobile applications in the future. Android is new—Android-powered devices first appeared on the scene in late 2008—but it has already grown tremendously, becoming the number one handset operating system in three short years.

And, most of all, thanks for your interest in this book! I sincerely hope you find it useful and at least occasionally entertaining.

## Prerequisites

If you are interested in programming for Android, you will need at least a basic understanding of how to program in Java. Android programming is done using Java syntax, plus a class library that resembles a subset of the Java SE library (plus Android-specific extensions). If you have not programmed in Java before, you probably should learn how that works before attempting to dive into programming for Android. The blog post <http://commonsware.com/blog/2010/08/02/java-good-parts-version.html> enumerates the various Java programming topics an Android developer needs to know. This subject is also dealt with by another Apress book, *Learn Java for Android Development*, by Jeff Friesen (Apress, 2010).

## Editions of This Book

This book is being produced via a partnership between Apress and CommonsWare. You are reading the Apress edition, which is available in print and in digital form from various digital book services, such as Safari.

CommonsWare continually updates the original material and makes it available to members of its Warescription program, under the title *The Busy Coder's Guide to Android Development*.

CommonsWare maintains a FAQ about this partnership at <http://commonsware.com/apress>.

## Source Code and Its License

The source code for this book is available at [www.apress.com](http://www.apress.com). All of the Android projects are licensed under the Apache 2.0 License, [www.apache.org/licenses/LICENSE-2.0.html](http://www.apache.org/licenses/LICENSE-2.0.html), in case you have the desire to reuse any of it.

Part



# Core Concept

# The Big Picture

Android is everywhere. Phones. Tablets. TVs and set-top boxes powered by Google TV. Soon, Android will be in cars, in in-flight entertainment systems on planes, and even in robots!

However, the general theme of Android devices will be smaller screens and/or no hardware keyboard. And, by the numbers, Android will probably be associated mostly with smartphones for the foreseeable future. For developers, this has both benefits and drawbacks, as described next. This chapter also describes the main components in an Android application and the Android features that you can exploit when developing your applications.

## Benefits and Drawbacks of Smartphone Programming

On the plus side, Android-style smartphones are sexy. Offering Internet services over mobile devices dates back to the mid-1990s and the Handheld Device Markup Language (HDML). However, only in recent years have phones capable of Internet access taken off. Now, thanks to trends like text messaging and products like Apple's iPhone, phones that can serve as Internet-access devices are rapidly gaining popularity. So, working on Android applications gives you experience with an interesting technology (Android) in a fast-moving market segment (Internet-enabled phones), which is always a good thing.

The problem comes when you actually have to program the darn things.

Anyone with experience in programming for PDAs or phones has felt the pain of phones simply being *small* in all sorts of dimensions:

- Screens are small (you will not get comments like, "Is that a 24-inch LCD in your pocket, or . . . ?").
- Keyboards, if they exist, are small.

- Pointing devices, if they exist, are annoying (as anyone who has lost their stylus will tell you) or inexact (large fingers and “multitouch” LCDs can sometimes be . . . problematic).
- CPU speed and memory are always behind what’s available on desktops and servers.

Moreover, applications running on a phone have to deal with the fact that they’re *on a phone*.

People with mobile phones tend to get very irritated when those phones do not work. Similarly, those same people will get irritated if your program “breaks” their phones by

- Tying up the CPU such that calls can’t be received.
- Not quietly fading into the background when a call comes in or needs to be placed, because the program doesn’t work properly with the rest of the phone’s operating system.
- Crashing the phone’s operating system, such as by leaking memory like a sieve.

Hence, developing programs for a phone is a different experience than developing desktop applications, web sites, or back-end server processes. The tools look different, the frameworks behave differently, and you have more limitations on what you can do with your programs.

What Android tries to do is meet you halfway:

- You get a commonly used programming language (Java) with some commonly used libraries (e.g., some Apache Commons APIs), with support for tools you may be used to using (Eclipse).
- You get a fairly rigid and uncommon framework in which your programs need to run so they can be “good citizens” on the phone and not interfere with other programs or the operation of the phone itself.

As you might expect, much of this book deals with that framework and how you write programs that work within its confines and take advantage of its capabilities.

## What Androids Are Made Of

When you write a desktop application, you are “master of your own domain.” You launch your main window and any child windows—like dialog boxes—that are needed. From your standpoint, you are your own world, leveraging features supported by the operating system, but largely ignorant of any other program that may be running on the computer at the same time. If you do interact with other programs, it is typically through an application programming interface (API), such as Java Database Connectivity (JDBC), or frameworks atop it, to communicate with MySQL or another database.



Android has similar concepts, but they are packaged differently and structured to make phones more crash-resistant:

- *Activities*: The building block of the user interface is the activity. You can think of an activity as being the Android analogue for the window or dialog box in a desktop application or the page in a classic web application. Android is designed to support lots of cheap activities, so you can allow users to keep tapping to open new activities and tapping the Back button to back up, just like they do in a web browser.
- *Services*: Activities are short-lived and can be shut down at any time. Services, on the other hand, are designed to keep running, if needed, independent of any activity, akin to the notion of services or daemons on other operating systems. You might use a service to check for updates to an RSS feed or to play back music even if the controlling activity is no longer operating. You will also use services for scheduled tasks (“cron jobs”) and for exposing custom APIs to other applications on the device, though those are relatively advanced capabilities.
- *Content providers*: Content providers provide a level of abstraction for any data stored on the device that is accessible by multiple applications. The Android development model encourages you to make your own data available to other applications, as well as your own applications. Building a content provider lets you do that, while maintaining complete control over how your data gets accessed. Content providers can be anything from web feeds, to local SQLite databases, and beyond.
- *Intents*: Intents are system messages that run around the inside of the device and notify applications of various events, from hardware state changes (e.g., an SD card was inserted), to incoming data (e.g., a Short Message Service [SMS] message arrived), to application events (e.g., your activity was launched from the device’s main menu). Intents are much like messages or events on other operating systems. Not only can you respond to an Intent, but you can create your own to launch other activities or to let you know when specific situations arise (e.g., raise such-and-so Intent when the user gets within 100 meters of this-and-such location).

## Stuff at Your Disposal

- *Storage*: You can package data files with your application for things that do not change, such as icons or help files. You also can carve out a small bit of space on the device itself, for databases or files containing user-entered or retrieved data needed by your application. And, if the user supplies bulk storage, like an SD card, you can read and write files on there as needed.

- *Network*: Android devices generally are Internet-ready, through one communications medium or another. You can take advantage of the Internet access at any level you wish, from raw Java sockets all the way up to a built-in WebKit-based web browser widget you can embed in your application.
- *Multimedia*: Android devices have the ability to play back and record audio and video. While the specifics may vary from device to device, you can query the device to learn its capabilities and then take advantage of the multimedia capabilities as you see fit, whether that is to play back music, take pictures with the camera, or use the microphone for audio note-taking.
- *Location services*: Android devices frequently have access to location providers, such as GPS and cell triangulation, which can tell your applications where the device is on the face of the Earth. In turn, you can display maps or otherwise take advantage of the location data, such as to track a device's movements if the device has been stolen.
- *Phone services*: Because Android devices are typically phones, your software can initiate calls, send and receive SMS messages, and do everything else you expect from a modern bit of telephony technology.

## The Big Picture...of This Book

Now that you have the Android big picture, here is what's coming in the rest of this book:

- The next two chapters are designed to get you going quickly with the Android environment, through a series of step-by-step, tutorial-style instructions for setting up the tools you need, creating your first project, and getting that first project running on the Android emulator.
- The three chapters that follow explain a bit more about what just happened in Chapters 2 and 3. We examine the Android project that we created, talk a bit more about Eclipse, and discuss some things we could add to the project to help it run on more devices and enhance its capabilities.
- The bulk of the book explores the various capabilities of the Android APIs—how to create components like activities, how to access the Internet and local databases, how to get your location and show it on a map, and so forth.

# How to Get Started

Without further ado, let's get you set up with the pieces and parts necessary to build an Android app.

**NOTE:** The instructions presented here are accurate as of the time of this writing. However, the tools change rapidly, so these instructions may be out of date by the time you read this. Please refer to the Android Developers web site for current instructions, using this as a base guideline of what to expect.

## Step 1: Set Up Java

When you write Android applications, you typically write them in Java source code. That Java source code is then turned into the stuff that Android actually runs (Dalvik bytecode in an Android package [APK] file).

Hence, the first thing you need to do is get set up with a Java development environment so that you are prepared to start writing Java classes.

### Install the JDK

You need to obtain and install the official Oracle Java SE Development Kit (JDK). You can obtain this from the Oracle Java web site for Windows and Linux, and from Apple for Mac OS X. The plain JDK (sans any “bundles”) should suffice. Follow the instructions supplied by Oracle or Apple for installing it on your machine. At the time of this writing, Android supports Java 5 and Java 6, with Java 7 likely to be supported by the time you are reading this.

## ALTERNATIVE JAVA COMPILERS

In principle, you are supposed to use the official Oracle JDK. In practice, it appears that OpenJDK also works, at least on Ubuntu. However, the further removed you get from the official Oracle implementation, the less likely it is that it will work. For example, the GNU Compiler for Java (GCJ) may not work with Android.

---

## Learn Java

This book, like most books and documentation on Android, assumes that you have basic Java programming experience. If you lack this, you really should consider spending a bit of time on Java fundamentals before you dive into Android. Otherwise, you may find the experience to be frustrating.

If you are in need of a crash course in Java to get involved in Android development, here are the concepts you need to learn, presented in no particular order:

- Language fundamentals (flow control, etc.)
- Classes and objects
- Methods and data members
- Public, private, and protected
- Static and instance scope
- Exceptions
- Threads and concurrency control
- Collections
- Generics
- File I/O
- Reflection
- Interfaces

One of the easiest ways of acquiring this knowledge is to read *Learn Java for Android Development* by Jeff Friesen (Apress, 2010).

## Step 2: Install the Android SDK

The Android SDK gives you all the tools you need to create and test Android applications. It comes in two parts: the base tools, and version-specific SDKs and related add-ons.

### Install the Base Tools

You can find the Android developer tools on the Android Developers web site at <http://developer.android.com>. Download the ZIP file that is appropriate for your platform and unzip it in a logical location on your machine—no specific path is required. Windows users also have the option of running a self-installing EXE file.

### Install the SDKs and Add-ons

Inside the `tools/` directory of your Android SDK installation from the previous step, you will see an `android` batch file or shell script. If you run that, you will be presented with the Android SDK and AVD Manager, shown in Figure 2–1.



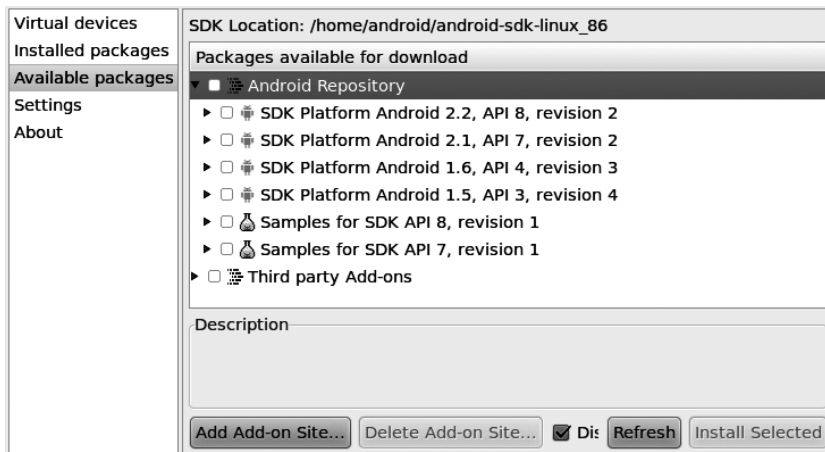
**Figure 2–1.** *Android SDK and AVD Manager*

At this point, you have some of the build tools, but you lack the Java files necessary to compile an Android application. You also lack a few additional build tools, and the files necessary to run an Android emulator. To address this, click the Available packages option on the left to open the screen shown in Figure 2–2.



**Figure 2–2.** *Android SDK and AVD Manager available packages*

Open the Android Repository branch of the tree. After a short pause, you will see a screen similar to Figure 2–3.



**Figure 2–3.** *Android SDK and AVD Manager available Android packages*

Check the boxes for the following items:

- “SDK Platform” for all Android SDK releases you want to test against
- “Documentation for Android SDK” for the latest Android SDK release
- “Samples for SDK” for the latest Android SDK release, and perhaps for older releases if you wish

Then, open the Third party Add-ons branch of the tree. After a short pause, you will see a screen similar to Figure 2–4.