



Beginning Android[®] Programming with Android Studio

J. F. DiMarzio

BEGINNING ANDROID® PROGRAMMING WITH ANDROID STUDIO

INTRODUCTION	xvii
CHAPTER 1 Getting Started with Android Programming.....	1
CHAPTER 2 Using Android Studio for Android Development	29
CHAPTER 3 Activities, Fragments, and Intents	47
CHAPTER 4 Getting to Know the Android User Interface	101
CHAPTER 5 Designing Your User Interface with Views.....	147
CHAPTER 6 Displaying Pictures and Menus with Views	203
CHAPTER 7 Data Persistence.....	231
CHAPTER 8 Content Providers	269
CHAPTER 9 Messaging	297
CHAPTER 10 Location-Based Services	325
CHAPTER 11 Networking.....	351
CHAPTER 12 Developing Android Services	381
APPENDIX Answers to Exercises	415
INDEX	421

BEGINNING

Android® Programming with Android Studio

J. F. DiMarzio



Beginning Android® Programming with Android Studio

Published by
John Wiley & Sons, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2017 by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-118-70559-9
ISBN: 978-1-118-70742-5 (ebk)
ISBN: 978-1-119-19609-9 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2016952673

Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Android is a registered trademark of Google, Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

*To my children Christian, Sophia, and Giovanni;
Katie, Sarah, and Joe; and my love Jennifer.*

CREDITS

Project Editor

Charlotte Kughen

Technical Editor

Chád (Shod) Darby

Production Editor

Athiyappan Lalith Kumar

Development Editor

Rick Kughen

Manager of Content Development and Assembly

Mary Beth Wakefield

Production Manager

Kathleen Wisor

Marketing Manager

Carrie Sherrill

Professional Technology and Strategy Director

Barry Pruett

Business Manager

Amy Knies

Executive Editor

Jim Minatel

Project Coordinator, Cover

Brent Savage

Proofreader

Nancy Bell

Indexer

Nancy Guenther

Cover Designer

Wiley

Cover Image

© iStockphoto.com/Leo Blanchette

ABOUT THE AUTHOR

J. F. DIMARZIO began developing computer programs—specifically games—in 1984 as a wide-eyed, curious child of the Atari age. Starting on the TRS-80 Color Computer II, in BASIC, he wrote several small text-based games and simple inventory applications. After leaving the Music program at the University of Massachusetts, J. F. moved into professional computer development; working for the United States Department of Defense, Walt Disney Imagineering, TechData, and The Walt Disney Company. In 2008, he started developing apps on the newly introduced Android platform (then on version 0.8). He has had 14 books published to date, 7 of which are on Android and Android game development.

ABOUT THE TECHNICAL EDITOR

CHÁD (SHOD) DARBY is an author, instructor, and speaker in the Java development world. As a recognized authority on Java applications and architectures, he has presented technical sessions at software development conferences worldwide (in the U.S., U.K., India, Italy, Russia, Netherlands, Singapore, Japan, and Australia). In his 20 years as a professional software architect, he's had the opportunity to work for Blue Cross/Blue Shield, Merck, Boeing, Red Hat, and a handful of startup companies.

Chád is a contributing author to several Java books, including *Professional Java E-Commerce* (Wrox Press), *Beginning Java Networking* (Wrox Press), and *XML and Web Services Unleashed* (Sams Publishing). Chád has Java certifications from Oracle and IBM. He holds a B.S. in Computer Science from Carnegie Mellon University.

ACKNOWLEDGMENTS

J. F. DIMARZIO would like to thank Charlotte Kughen, Jim Minatel, Rick Kughen, and the team at John Wiley & Sons publishing; and Stacey Czarnowski, Katrina Bevan, and the team at Studio B.

CONTENTS

INTRODUCTION

xvii

CHAPTER 1: GETTING STARTED WITH ANDROID PROGRAMMING 1

What Is Android?	2
Android Versions	2
Features of Android	3
Architecture of Android	4
Android Devices in the Market	4
The Android Market	7
Obtaining the Required Tools	8
Android Studio	8
Android SDK	14
Creating Android Virtual Devices (AVDs)	15
The Android Developer Community	24
Launching Your First Android Application	24
Summary	26

CHAPTER 2: USING ANDROID STUDIO FOR ANDROID DEVELOPMENT 29

Exploring the IDE	30
Using Code Completion	38
Debugging Your Application	39
Setting Breakpoints	39
Navigating Paused Code	43
Publishing Your Application	44
Generating a Signed APK	44
Summary	45

CHAPTER 3: ACTIVITIES, FRAGMENTS, AND INTENTS 47

Understanding Activities	48
Applying Styles and Themes to an Activity	53
Hiding the Activity Title	54
Displaying a Dialog Window	56
Displaying a Progress Dialog	59

Linking Activities Using Intents	61
Returning Results from an Intent	66
Passing Data Using an Intent Object	70
Fragments	75
Adding Fragments Dynamically	81
Life Cycle of a Fragment	84
Interactions Between Fragments	88
Understanding the Intent Object	92
Using Intent Filters	93
Displaying Notifications	94
Summary	98
CHAPTER 4: GETTING TO KNOW THE ANDROID USER INTERFACE	101
Understanding the Components of a Screen	102
Views and ViewGroups	103
FrameLayout	104
LinearLayout (Horizontal) and LinearLayout (Vertical)	104
TableLayout	114
RelativeLayout	116
FrameLayout	117
ScrollView	121
Adapting to Display Orientation	124
Anchoring Views	125
Managing Changes to Screen Orientation	127
Persisting State Information During Changes in Configuration	131
Detecting Orientation Changes	132
Controlling the Orientation of the Activity	132
Utilizing the Action Bar	133
Adding Action Items to the Action Bar	136
Creating the User Interface Programmatically	142
Listening for UI Notifications	145
Summary	145
CHAPTER 5: DESIGNING YOUR USER INTERFACE WITH VIEWS	147
Using Basic Views	148
TextView View	148
Button, ImageButton, EditText, CheckBox, ToggleButton, RadioButton, and RadioGroup Views	149

ProgressBar View	158
AutoCompleteTextView View	164
Using Picker Views	167
TimePicker View	167
DatePicker View	172
Using List Views to Display Long Lists	175
ListView View	175
Using the Spinner View	184
Understanding Specialized Fragments	186
Using a ListFragment	187
Using a DialogFragment	191
Using a PreferenceFragment	194
Summary	199
CHAPTER 6: DISPLAYING PICTURES AND MENUS WITH VIEWS	203
Using Image Views to Display Pictures	203
ImageView View	204
ImageSwitcher	206
GridView	211
Using Menus with Views	215
Creating the Helper Methods	215
Options Menu	217
Context Menu	220
Using WebView	225
WebView	225
Summary	228
CHAPTER 7: DATA PERSISTENCE	231
Saving and Loading User Preferences	232
Accessing Preferences Using an Activity	232
Programmatically Retrieving and Modifying the Preferences Values	242
Persisting Data to Files	246
Saving to Internal Storage	246
Saving to External Storage (SD Card)	250
Choosing the Best Storage Option	253
Creating and Using Databases	254
Creating the DBAdapter Helper Class	254
Using the Database Programmatically	260
Summary	266

CHAPTER 8: CONTENT PROVIDERS	269
Sharing Data in Android	270
Using a Content Provider	271
Predefined Query String Constants	278
Projections	279
Filtering	280
Sorting	280
Creating Your Own Content Providers	281
Using the Content Provider	289
Summary	294
CHAPTER 9: MESSAGING	297
SMS Messaging	298
Sending SMS Messages Programmatically	298
Sending SMS Messages Using Intent	302
Receiving SMS Messages	302
Caveats and Warnings	318
Sending Email	319
Summary	322
CHAPTER 10: LOCATION-BASED SERVICES	325
Displaying Maps	326
Creating the Project	326
Obtaining the Maps API Key	326
Displaying the Map	327
Displaying the Zoom Control	329
Changing Views	332
Navigating to a Specific Location	333
Getting the Location That Was Touched	335
Geocoding and Reverse Geocoding	337
Getting Location Data	340
Monitoring a Location	348
Summary	348
CHAPTER 11: NETWORKING	351
Consuming Web Services Using HTTP	352
Downloading Binary Data	354
Downloading Text Content	361
Accessing Web Services Using the GET Method	364

Consuming JSON Services	370
Summary	377
CHAPTER 12: DEVELOPING ANDROID SERVICES	381
<hr/>	
Creating Your Own Services	382
Performing Long-Running Tasks in a Service	386
Performing Repeated Tasks in a Service	391
Executing Asynchronous Tasks on Separate Threads Using IntentService	394
Establishing Communication Between a Service and an Activity	397
Binding Activities to Services	401
Understanding Threading	406
Summary	411
APPENDIX: ANSWERS TO EXERCISES	415
<hr/>	
<i>INDEX</i>	421

INTRODUCTION

I first started playing with the Android Software Development Kit (SDK) before it was officially released as version 0.8. Back then, the tools were unpolished, the application programming interfaces (APIs) in the SDK were unstable, and the documentation was sparse. Android is now one of the most popular mobile operating systems in the world.

Every day, more developers are trying Android development using Android Studio, either as a hobby or professionally. The current challenge for many beginning developers who want to move into the exciting world of android development is trying to determine where to start.

It was with this challenge in mind that I was motivated to write this book, one that could benefit beginning Android programmers and enable them to write progressively more sophisticated applications.

This book is written to help jump-start beginning Android developers, covering the necessary topics in a linear manner so that you can build on your knowledge without being overwhelmed by the details. I adopt the philosophy that the best way to learn is by doing. The numerous Try It Out sections in each chapter first show you how to build something. Then the follow-up How It Works sections explain how everything works. I have also taken this opportunity to further improve the previous edition of this book by addressing feedback from readers and adding additional topics that are important to beginning Android developers.

Although Android programming is a huge topic, my aim for this book is threefold: to get you started with the fundamentals, to help you understand the underlying architecture of the SDK, and to appreciate why things are done in certain ways. It is beyond the scope of any book to cover everything under the sun related to Android programming. However, I am confident that after reading this book (and doing the exercises), you will be well equipped to tackle your next Android programming challenge.

WHO THIS BOOK IS FOR

This book is targeted for the beginning Android developer who wants to start developing applications using Google's Android SDK. To truly benefit from this book, you should have some background in programming and at least be familiar with object-oriented programming (OOP) concepts. If you are totally new to Java—the language used for Android development—you might want to take a programming course in Java programming or grab one of many good books on Java programming. In my experience, if you already know C# or VB.NET, learning Java doesn't require too much effort. If you're already comfortable with C# or VB.NET, you should be comfortable just following along with the Try It Outs.

For those totally new to programming, I know the lure of developing mobile apps and making some money is tempting. However, before attempting to try out the examples in this book, I think a better starting point would be to first learn the basics of programming.

NOTE *All the examples discussed in this book were written and tested using version N (Nougat) of the Android SDK on Android Studio 2 previews (1 through 6). Although every effort is made to ensure that all the tools used in this book are the latest, it is always possible that by the time you read this book a newer version of the tools will be available. If so, some of the instructions and/or screenshots may differ slightly. However, any variations should be manageable.*

WHAT THIS BOOK COVERS

This book covers the fundamentals of Android programming using the Android SDK. It is divided into 12 chapters and one appendix.

- **Chapter 1: Getting Started with Android Programming** covers the basics of the Android OS and its current state. You are introduced to the features of Android devices, as well as some of the popular devices on the market. You also find out how to download and install Android Studio to develop Android applications, and then you see how to test them on the Android emulator.
- **Chapter 2: Using Android Studio for Android Development** walks you through many of the different elements within Android Studio. You are introduced to the IDE (Integrated Development Environment) and its pieces. Finally, you discover how to publish a finished application
- **Chapter 3: Activities, Fragments, and Intents** gets you acquainted with these three fundamental concepts in Android programming. Activities and fragments are the building blocks of an Android application. You find out how to link activities to form a complete Android application using intents, one of the unique characteristics of the Android OS.
- **Chapter 4: Getting to Know the Android User Interface** covers the various components that make up the user interface (UI) of an Android application. You are introduced to the various layouts you can use to build the UI of your application. You also learn about the numerous events that are associated with the UI when users interact with the application.
- **Chapter 5: Designing Your User Interface with Views** walks you through the various basic views you can use to build your Android UI. You learn three main groups of views: basic views, picker views, and list views. You also find out about the specialized fragments available in Android 3.0 and 4.0.

- **Chapter 6: Displaying Pictures and Menus with Views** continues the exploration of views. Here, you see how to display images using the various image views, as well as display options and context menus in your application. This chapter ends with some additional cool views that you can use to spice up your application.
- **Chapter 7: Data Persistence** shows you how to save, or store, data in your Android application. In addition to being introduced to the various techniques to store user data, you also find out about file manipulation and how to save files onto internal and external storage (SD card). In addition, you learn how to create and use a SQLite database in your Android application.
- **Chapter 8: Content Providers** discusses how data can be shared among different applications on an Android device. You see how to use a content provider and then build one yourself.
- **Chapter 9: Messaging** explores two of the most interesting topics in mobile programming—sending SMS messages and email. You learn how to programmatically send and receive SMS and email messages, as well as how to intercept incoming SMS messages so that the built-in Messaging application is not able to receive any messages.
- **Chapter 10: Location-Based Services** demonstrates how to build a location-based service application using Google Maps. You also find out how to obtain geographical location data and then display the location on the map.
- **Chapter 11: Networking** explores how to connect to web servers to download data. You see how XML and JSON web services can be consumed in an Android application. This chapter also explains sockets programming, and you see how to build a chat client in Android.
- **Chapter 12: Developing Android Services** demonstrates how you can write applications using services. Services are background applications that run without a UI. You learn how to run your services asynchronously on a separate thread, and how your activities can communicate with them.
- **Appendix: Answers to Exercises** contains the solutions to the end-of-chapter exercises found in every chapter.

HOW THIS BOOK IS STRUCTURED

This book breaks down the task of learning Android programming into several smaller chunks, enabling you to digest each topic before delving into a more advanced one.

If you are a total beginner to Android programming, start with Chapter 1. After you have familiarized yourself with the basics, head to Chapter 2 and get to know the Android Studio IDE. When you are ready, continue with Chapter 3 and gradually move into more advanced topics.

A feature of this book is that all the code samples in each chapter are independent of those discussed in previous chapters. This gives you the flexibility to dive into the topics that interest you and start working on the Try It Out projects.

WHAT YOU NEED TO USE THIS BOOK

All the examples in this book run on the Android emulator (which is included as part of the Android SDK and Android Studio). However, to get the most out of this book, it would be useful to have a real Android device (though it's not absolutely necessary).

CONVENTIONS

To help you get the most from the text and keep track of what's happening, a number of conventions are used throughout the book.

TRY IT OUT These Are Exercises or Examples for You to Follow

The Try It Out sections appear once or more per chapter. These are exercises to work through as you follow the related discussion in the text.

1. They consist of a set of numbered steps.
2. Follow the steps with your copy of the project files.

How It Works

After each Try It Out, the code you've typed is explained in detail.

As for other conventions in the text:

- New terms and important words are *highlighted* in italic when first introduced.
- Keyboard combinations are treated like this: Ctrl+R.
- Filenames, URLs, and code within the text are treated like so: `persistence.properties`.
- Code is presented in two different ways:

We use a monofont type with no highlighting for most code examples.

We use **bolding** to emphasize code that is of particular importance in the present context.

NOTE Notes, tips, hints, tricks, and asides to the current discussion look like this.

SOURCE CODE

As you work through the examples in this book, you may choose either to type in all the code manually or to use the source code files that accompany the book. All the source code used in this book is available for download at www.wrox.com. When at the site, simply locate the book's title (use the Search box or one of the title lists) and click the Download Code link on the book's detail page to obtain all the source code for the book.

You'll find the filename of the project you need at the end of the title of the Try it Out features:

TRY IT OUT Understanding the Life Cycle of an Activity (Activity101.zip)

After you download the code, decompress it with your favorite compression tool. Alternatively, go to the main Wrox code download page at www.wrox.com/dynamic/books/download.aspx to see the code available for this book as well as for all other Wrox books.

NOTE *Because many books have similar titles, you might find it easiest to search by ISBN; this book's ISBN is 978-1-118-70559-9.*

ERRATA

We make every effort to ensure that there are no errors in the text or in the code. However, no one is perfect, and mistakes do occur. If you find an error in one of our books, such as a spelling mistake or faulty piece of code, we would be very grateful for your feedback. By sending in errata, you might save another reader hours of frustration and at the same time help us provide even higher-quality information.

To find the errata page for this book, go to www.wrox.com and locate the title using the Search box or one of the title lists. Then, on the book details page, click the Book Errata link. On this page, you can view all errata that has been submitted for this book and posted by Wrox editors.

NOTE *A complete book list, including links to each book's errata, is also available at www.wrox.com/misc-pages/booklist.shtml.*

If you don't spot "your" error on the Book Errata page, go to www.wrox.com/contact/techsupport.shtml and complete the form there to send us the error you have found. We'll check the information and, if appropriate, post a message to the book's errata page and fix the problem in subsequent editions of the book.

P2P.WROX.COM

For author and peer discussion, join the P2P forums at p2p.wrox.com. The forums are a web-based system for you to post messages relating to Wrox books and related technologies. There, you also can interact with other readers and technology users. The forums offer a subscription feature that enables you to receive emails about topics of interest (of your choosing) when new posts are made to the forums. Wrox authors, editors, other industry experts, and your fellow readers are present on these forums.

At p2p.wrox.com, you will find a number of different forums that will help you not only as you read this book but also as you develop your own applications. To join the forums, follow these steps:

1. Go to p2p.wrox.com and click the Register link.
2. Read the terms of use and click Agree.
3. Complete the required information to join as well as any optional information you want to provide and click Submit.
4. You will receive an email with information describing how to verify your account and complete the joining process.

NOTE *You can read messages in the forums without joining P2P, but in order to post your own messages, you must join.*

After you join, you can post new messages and respond to messages posted by other users. You can read messages at any time on the web. If you want to have new messages from a particular forum emailed to you, click the Subscribe to This Forum icon next to the forum name in the forum listing.

For more information about how to use the Wrox P2P, be sure to read the P2P FAQs for answers to questions about how the forum software works, as well as many common questions specific to P2P and Wrox books. To read the FAQs, click the FAQ link on any P2P page.

1

Getting Started with Android Programming

WHAT YOU WILL LEARN IN THIS CHAPTER

- What is Android?
- Android versions and its feature set
- The Android architecture
- The various Android devices on the market
- The Android Market application store
- How to obtain the tools and SDK for developing Android applications
- How to develop your first Android application

CODE DOWNLOAD *There are no code downloads for this chapter.*

Welcome to the world of Android! This chapter explains what Android is and what makes it so compelling to both developers and device manufacturers. It also shows you how to obtain and set up all the necessary tools so that you can test your application on an Android emulator in Android Studio 2 and how to get started with developing your first Android application. By the end of this chapter, you will be equipped with the basic knowledge you need to explore more sophisticated techniques and tricks for developing your next killer Android application.

WHAT IS ANDROID?

Android is a mobile operating system that is based on a modified version of Linux. It was originally developed by a startup of the same name, Android, Inc. In 2005, as part of its strategy to enter the mobile space, Google purchased Android, Inc. and took over its development work (as well as its development team).

Google wanted the Android OS to be open and free, so most of the Android code was released under the open source Apache License. That means anyone who wants to use Android can do so by downloading the full Android source code. Moreover, vendors (typically hardware manufacturers) can add their own proprietary extensions to Android and customize Android to differentiate their products from others. This development model makes Android very attractive to vendors, especially those companies affected by the phenomenon of Apple’s iPhone, which was a hugely successful product that revolutionized the smartphone industry. When the iPhone was launched, many smartphone manufacturers had to scramble to find new ways of revitalizing their products. These manufacturers saw Android as a solution, meaning they will continue to design their own hardware and use Android as the operating system that powers it. Some companies that have taken advantage of Android’s open source policy include Motorola and Sony Ericsson, which have been developing their own mobile operating systems for many years.

The main advantage to adopting Android is that it offers a unified approach to application development. Developers need only develop for Android in general, and their applications should be able to run on numerous different devices, as long as the devices are powered using Android. In the world of smartphones, applications are the most important part of the success chain.

Android Versions

Android has gone through quite a number of updates since its first release. Table 1-1 shows the various versions of Android and their codenames.

TABLE 1-1: A Brief History of Android Versions

ANDROID VERSION	RELEASE DATE	CODENAME
1.1	February 9, 2009	
1.5	April 30, 2009	Cupcake
1.6	September 15, 2009	Donut
2.0/2.1	October 26, 2009	Éclair
2.2	May 20, 2010	Froyo
2.3	December 6, 2010	Gingerbread
3.0/3.1/3.2	February 22, 2011	Honeycomb
4.0	October 18, 2011	Ice Cream Sandwich

ANDROID VERSION	RELEASE DATE	CODENAME
4.1	July 9, 2012	Jelly Bean
4.4	October 31, 2013	KitKat
5.0	November 12, 2014	Lollipop
6.0	October 5, 2015	Marshmallow
7.0	TBD	Nougat

In 2016, Google released Android 7.0; the following are the key changes in Android 7.0:

- Split-screen multi-window mode
- Redesigned notification shade
- Refined “Doze” feature
- Switch from JRE (Java Runtime Environment) to OpenJDK

One important thing to keep in mind as you are looking at Android versions is that each version has its own features and APIs (application programming interfaces). Therefore, if your application is written for the newest version of Android, and it uses an API that was not present in an older version of Android, then only devices running that newer version of Android will be able to use your application.

Features of Android

Because Android is open source and freely available to manufacturers for customization, there are no fixed hardware or software configurations. However, the base Android OS supports many features, including

- **Storage**—SQLite, a lightweight relational database, for data storage. Chapter 7 discusses data storage in more detail.
- **Connectivity**—GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth (includes A2DP and AVRCP), Wi-Fi, LTE, and WiMAX. Chapter 11 discusses networking in more detail.
- **Messaging**—Both SMS and MMS. Chapter 9 discusses messaging in more detail.
- **Media support** H.263, H.264 (in 3GP or MP4 container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), AAC, HE-AAC (in MP4 or 3GP container), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP.
- **Hardware support**—Accelerometer sensor, camera, digital compass, proximity sensor, and GPS.
- **Multi-touch**—Multi-touch screens.
- **Multi-tasking**—Multi-tasking applications.
- **Tethering**—Sharing of Internet connections as a wired/wireless hotspot.

Android’s web browser is based on the open source WebKit and Chrome’s V8 JavaScript engine.

Architecture of Android

To understand how Android works, take a look at Figure 1-1, which shows the various layers that make up the Android operating system (OS).

The Android OS is roughly divided into five sections in four main layers:

- **Linux kernel**—This is the kernel on which Android is based. This layer contains all the low-level device drivers for the various hardware components of an Android device.
- **Libraries**—These contain the code that provides the main features of an Android OS. For example, the SQLite library provides database support so that an application can use it for data storage. The WebKit library provides functionalities for web browsing.
- **Android runtime**—The Android runtime is located in the same layer with the libraries and provides a set of core libraries that enable developers to write Android apps using the Java programming language. The Android runtime also includes the Dalvik virtual machine, which enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine. (Android applications are compiled into Dalvik executables). Dalvik is a specialized virtual machine designed specifically for Android and optimized for battery-powered mobile devices with limited memory and CPU power.
- **Application framework**—The application framework exposes the various capabilities of the Android OS to application developers so that they can make use of them in their applications.
- **Applications**—At this top layer are the applications that ship with the Android device (such as Phone, Contacts, Browser, and so on), as well as applications that you download and install from the Android Market. Any applications that you write are located at this layer.

Android Devices in the Market

Android devices come in all shapes and sizes including, but not limited to, the following types of devices:

- Smartphones
- Tablets
- E-reader devices
- Internet TVs
- Automobiles
- Smartwatches

Chances are good that you own at least one of the preceding devices. Figure 1-2 shows the Samsung Galaxy Edge 7.

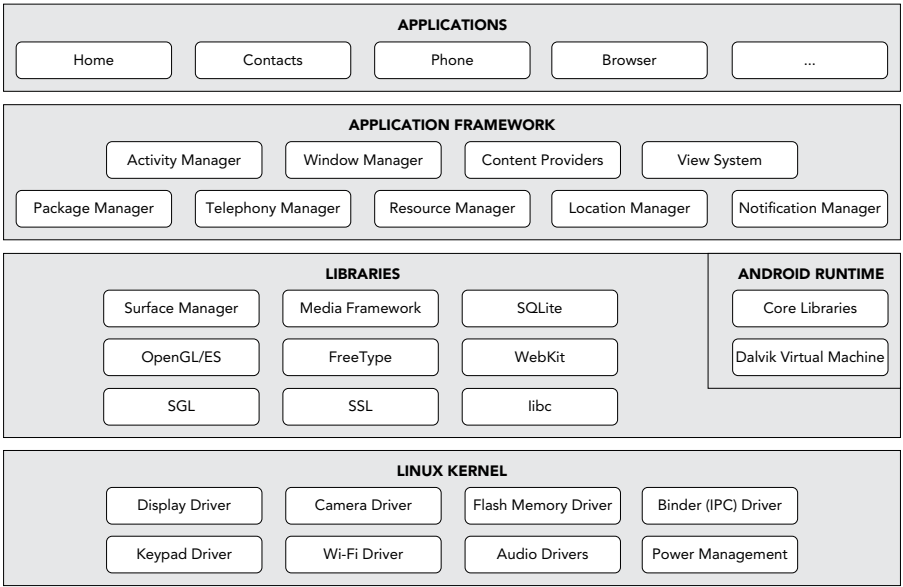


FIGURE 1-1

**FIGURE 1-2**

Another popular category of devices is the tablet. Tablets typically come in two sizes: 7" and 10", measured diagonally.

Besides smartphones and tablets, Android is used in dedicated devices, such as e-book readers. Figure 1-4 shows the Barnes and Noble's NOOK Color running the Android OS.

In addition to the popular mobile devices I've already mentioned, Android is finding its way onto your wrist. Smartwatches, and "wearables" in general, have become a major segment of the Android population. Figure 1-3 shows the Motorola Moto 360 Smartwatch, which runs Android Wear (a version of Android OS specifically designed for wearables).

At the time of writing, the Samsung Galaxy Nexus (see Figure 1-4) is the only device running a pure version of Android. Many manufacturers add their own modifications to the Android OS for use on their specific devices. Motorola devices

**FIGURE 1-3**

have Motoblur, HTC devices have HTC Sense, and so on. However, the Nexus devices always run a clean version of Android with no modifications.

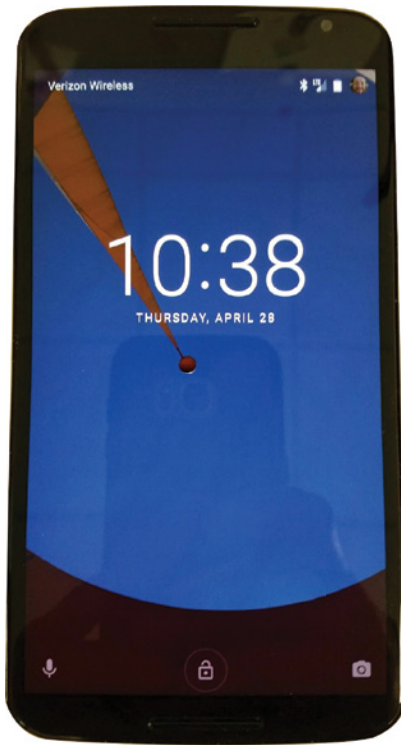


FIGURE 1-4

The Android Market

As mentioned earlier, one of the main factors determining the success of a smartphone platform is the applications that support it. It is clear from the success of the iPhone that applications play a very vital role in determining whether a new platform swims or sinks. Also, making these applications accessible to the general user is extremely important.

Users can simply use the Google Play application that is preinstalled on their Android devices to directly download third-party applications to their devices. Both paid and free applications are available in the Google Play Store, although paid applications are available only to users in certain countries because of legal issues.

NOTE Chapter 13 discusses more about Google Play Store and how you can sell your own applications in it.

OBTAINING THE REQUIRED TOOLS

Now that you know what Android is and what its feature set contains, you are probably anxious to get your hands dirty and start writing some applications! Before you write your first app, however, you need to download the required tools.

For Android development, you can use a Mac, a Windows PC, or a Linux machine. You can freely download all the necessary tools. Most of the examples provided in this book are written to work on Android Studio. For this book, I am using a Windows 10 computer to demonstrate all the code samples. If you are using a Mac or Linux computer, the screenshots should look similar. Some minor differences might be present, but you should be able to follow along without problems.

Let the fun begin!

JAVA JDK 8

The Android Studio 2 makes use of the Java SE Development Kit 8 (JDK). If your computer does not have the JDK 8 installed, you should start by downloading it from www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html and installing it prior to moving to the next section.

Android Studio

The first and most important piece of software you need to download is Android Studio 2. After you have downloaded and installed Android Studio 2, you can use the SDK Manager to download and install multiple versions of the Android SDK. Having multiple versions of the SDK available enables you to write programs that target different devices. For example, you can write one version of an application that specifically targets Android Nougat, but because that flavor of Android is on less than 1% of devices, with multiple versions of the SDK you can also write a version of your app that uses older features and targets Marshmallow or Lollipop users. You can use the Android Device Manager to set up device emulators.

You can download Android Studio 2 from <http://developer.android.com/sdk/index.html> (see Figure 1-5).

Android Studio 2 is packaged in an executable. Run the install process to set up Android Studio 2. After you've downloaded and run the setup executable, use the following steps to go through the installation process:

1. Accept the terms and conditions shown in Figure 1-6.
2. If you have an older version of Android Studio already installed on your computer, the Android Studio Setup prompts you to automatically uninstall it. Even though the old version of Android Studio will be uninstalled, the settings and configurations are retained. You have an opportunity to reapply those settings and configurations to Android Studio 2 after the setup has completed. Figure 1-7 shows the screen where you are prompted to uninstall an old version of Android Studio.

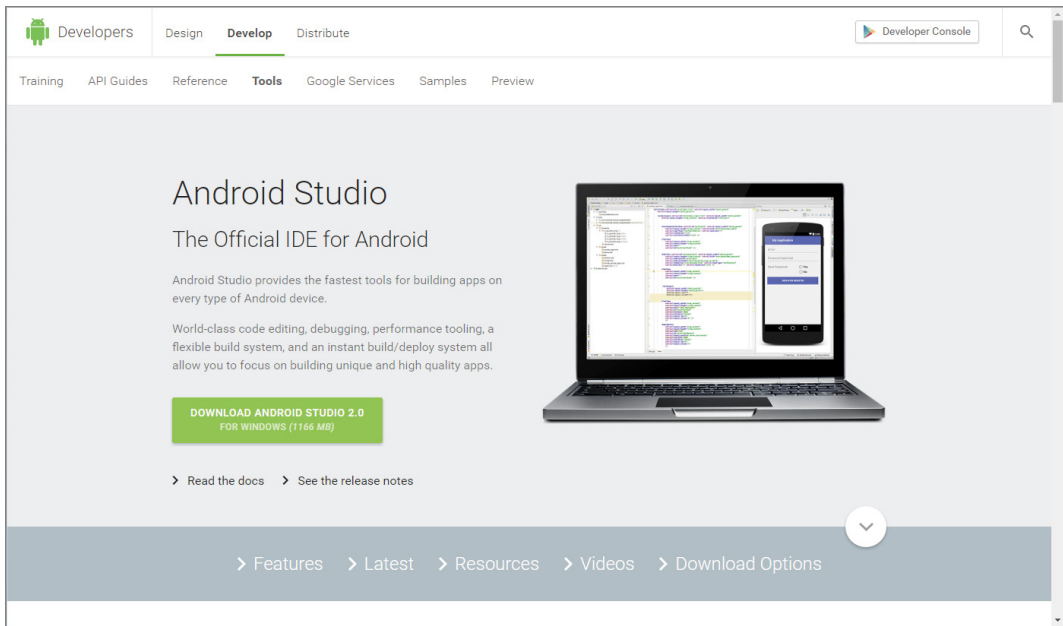


FIGURE 1-5

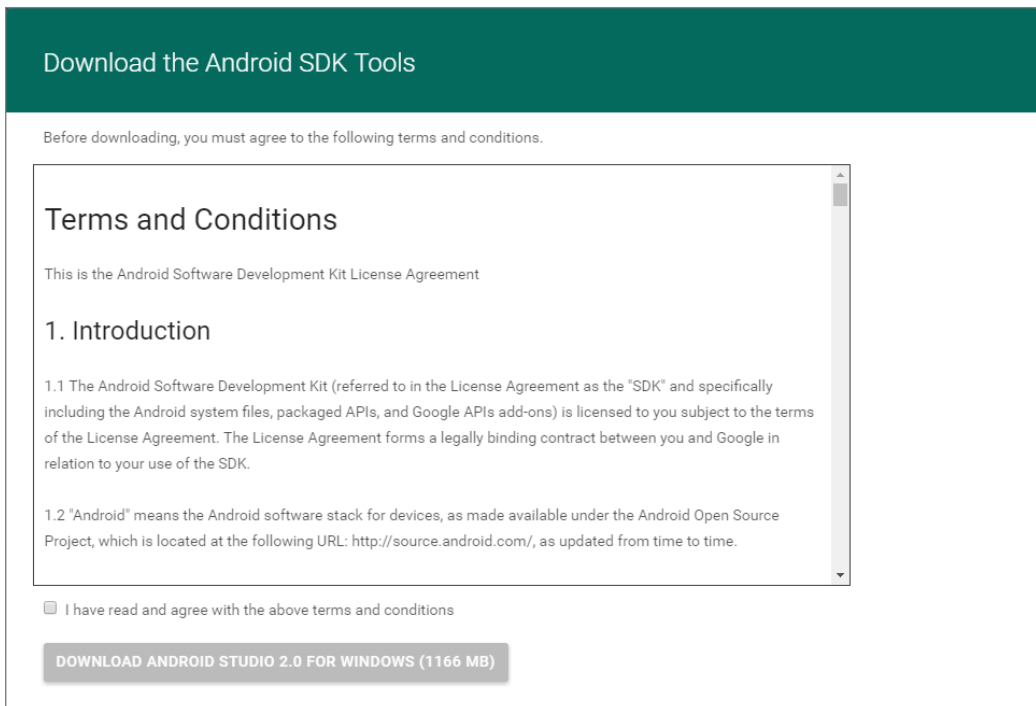


FIGURE 1-6

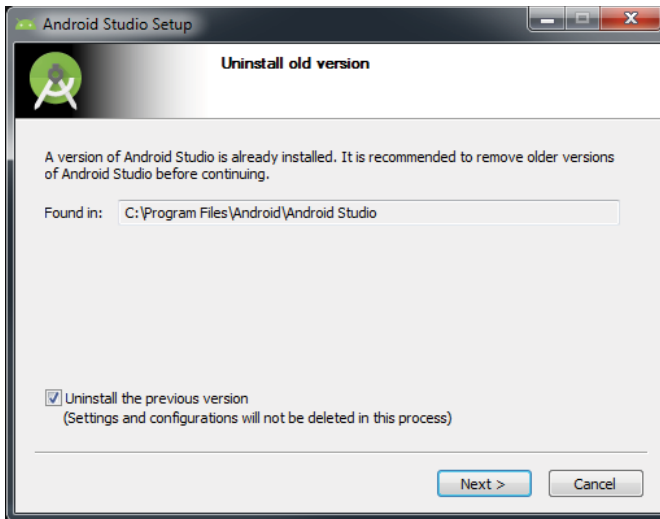


FIGURE 1-7

3. Click Next on the Welcome to Android Studio Setup screen (see Figure 1-8).



FIGURE 1-8

4. Pick which components of Android Studio you want to install from the screen shown in Figure 1-9. Android Studio is selected by default (and cannot be deselected), which makes sense given that you are going through all of this trouble for the distinct purpose of installing Android Studio. Android SDK and Android Virtual Device are also selected by default. Click Next to accept the default choices and continue.

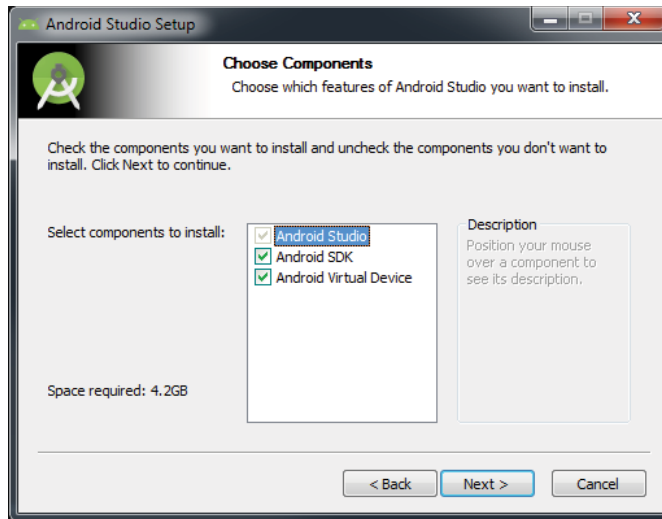


FIGURE 1-9

5. You are presented with the License Agreement, as shown in Figure 1-10. Click I Agree to continue.

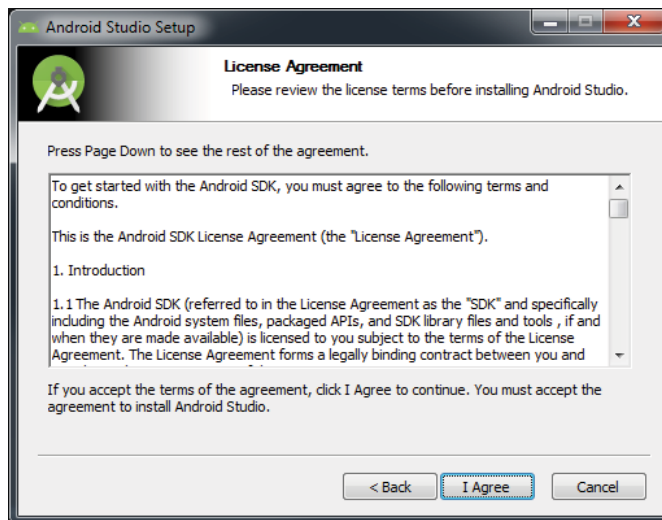


FIGURE 1-10

6. On the configuration settings screen, it is best to accept the default locations specified by the setup process and click Next to continue. You see the Choose Start Menu Folder screen (shown in Figure 1-11). Click Install to kick off the Android Studio 2 installation.

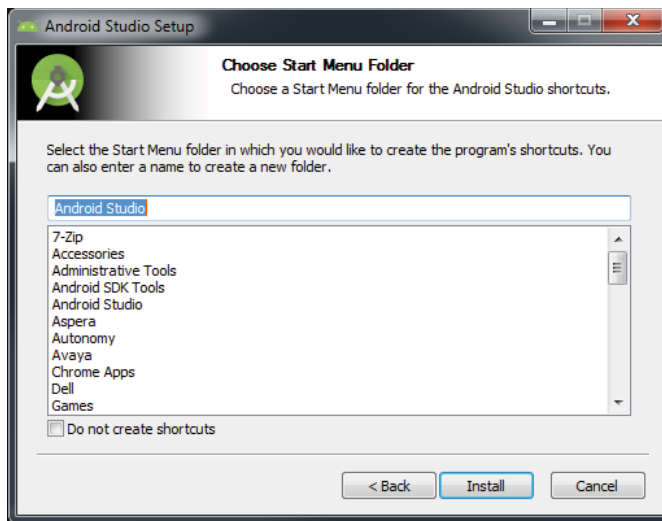


FIGURE 1-11

7. Installing Android Studio 2 could take a few minutes, depending on the speed of your computer. You are presented with a progress bar to help you track the state of the installation. Android Studio 2 is installed with a default SDK (Software Development Kit), in this case Marshmallow. Later in the process you have the opportunity to install other SDKs. The Android SDK allows you to develop and write applications geared for a specific version of Android. In other words, applications written with the Marshmallow SDK run on Android devices running Marshmallow, but they also possibly run on other versions depending on which features of the SDK you used in the application.
8. When the install is complete, you will see a Completing Android Studio Setup screen (shown in Figure 1-12). Leave the Start Android Studio box checked and click Finish.



FIGURE 1-12

9. Android Studio 2 prompts you to either import settings from a previous version of Android Studio or continue with new settings. If you uninstalled a previous version in the first step of the installation process, Android Studio offers you a chance to recover the settings used in that previous version and apply them to Android Studio 2 (see Figure 1-13).

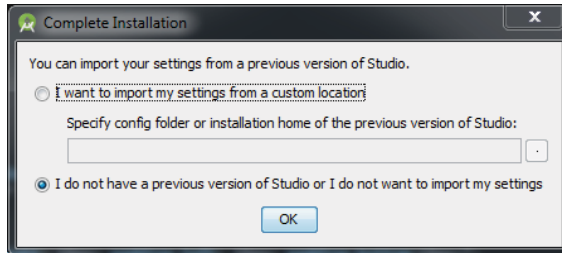


FIGURE 1-13

Now that Android Studio 2 is installed, you need to adjust the settings and options using the following steps:

1. Click Continue at the Welcome screen and choose Standard from the Install Type selection screen shown in Figure 1-14. Click Next to continue.

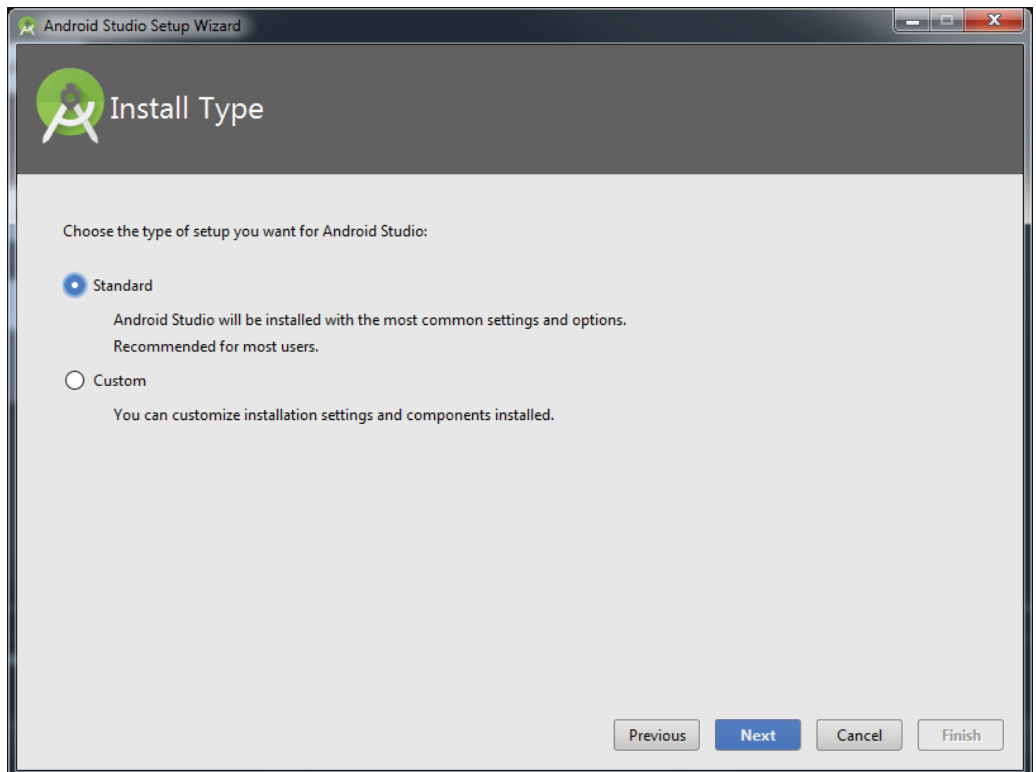


FIGURE 1-14

2. Click Finish on the Verify Settings screen, and Android Studio 2 finalizes the setup process. You know the process is complete when you are greeted with the Welcome to Android Studio screen (see Figure 1-15).

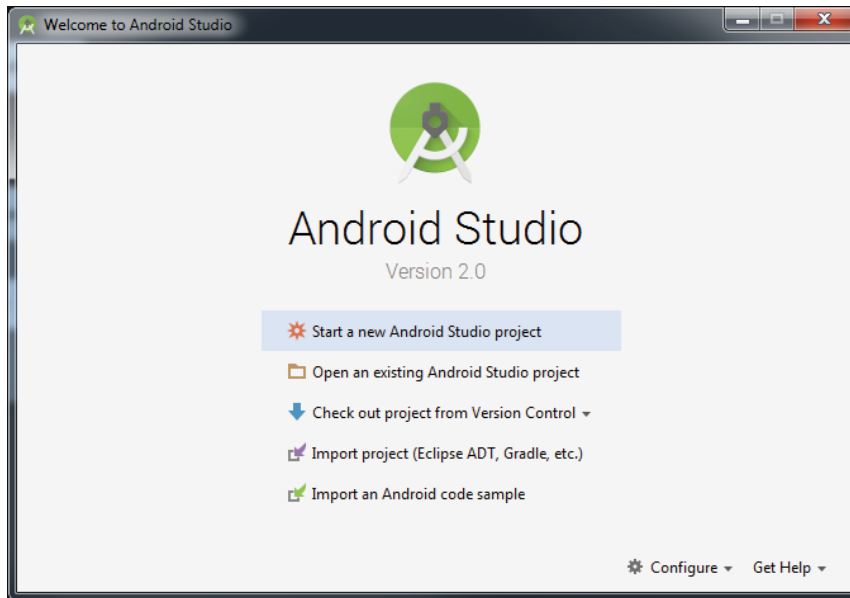


FIGURE 1-15

Now that Android Studio is set up, it's time to install the latest and greatest Android SDK.

Android SDK

The most important piece of software you need to download is, of course, the Android SDK. The Android SDK contains all of the packages and tools required to develop a functional Android application. The SDKs are named after the version of Android OS to which they correspond. By default, the Marshmallow SDK was installed with Android Studio 2, which means you can develop applications that will run seamlessly on devices with Android Marshmallow.

However, if you want to install a different Android SDK, you can do so using the SDK Manager from the Android Studio welcome screen (shown in Figure 1-15). From this screen, click the Configure drop-down menu in the lower-right corner. The Configure selection menu opens. Choose SDK Manager from this menu.

The SDK configuration screen, shown in Figure 1-16, shows that the Marshmallow SDK is already installed. Android N is available to be installed (as of the writing of this book Android Nougat was in a finalized beta, so it might be named differently now).

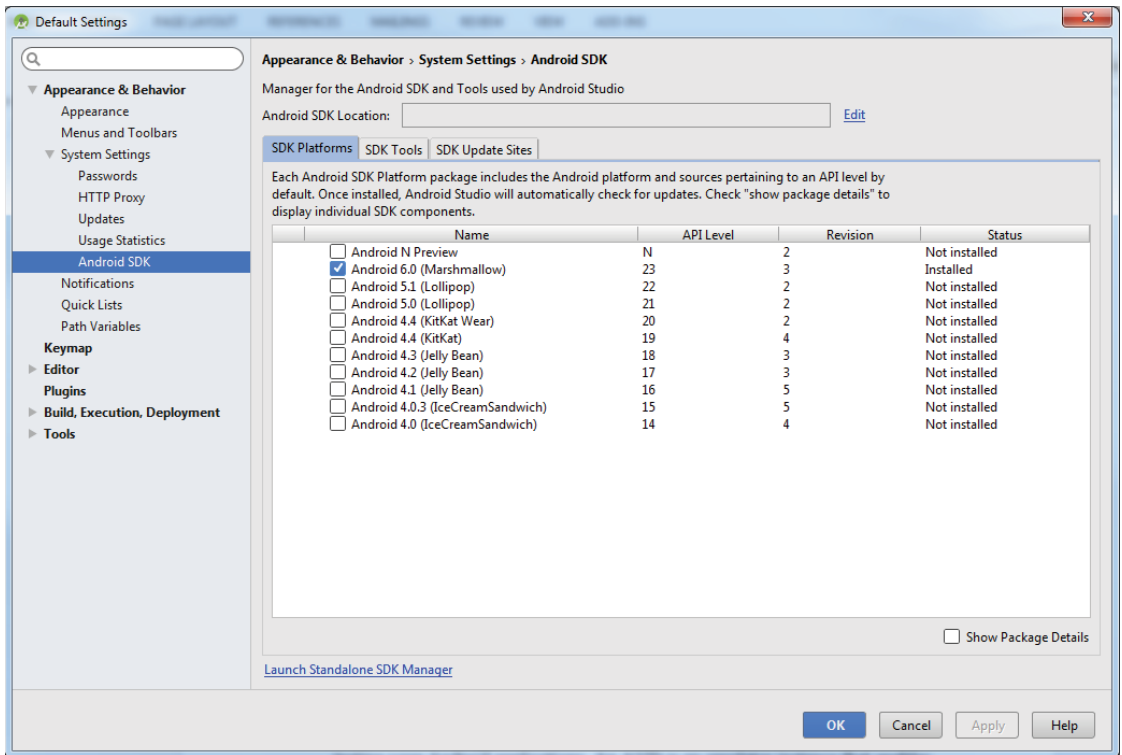


FIGURE 1-16

Select Android Nougat, click Apply, and then click OK. However, before the SDK is installed you must accept the licensing agreement as shown in Figure 1-17.

The setup process for Android Studio is now complete. The next section explains how to set up an Android Virtual Device that you can use to test your applications.

Creating Android Virtual Devices (AVDs)

The next step is to create an Android Virtual Device (AVD) you can use for testing your Android applications. An AVD is an emulator instance that enables you to model an actual device. Each AVD consists of a hardware profile; a mapping to a system image; and emulated storage, such as a secure digital (SD) card. One important thing to remember about emulators is that they are not perfect. There are some applications, such as games (which are GPU heavy) or applications that use sensors such as the GPS or accelerometer. These types of applications cannot be simulated with the same speed or consistency within an emulator as they can when running on an actual device. However, the emulator is good for doing some generalized testing of your applications.

You can create as many AVDs as you want to test your applications with different configurations. This testing is important to confirm the behavior of your application when it is run on different devices with varying capabilities.

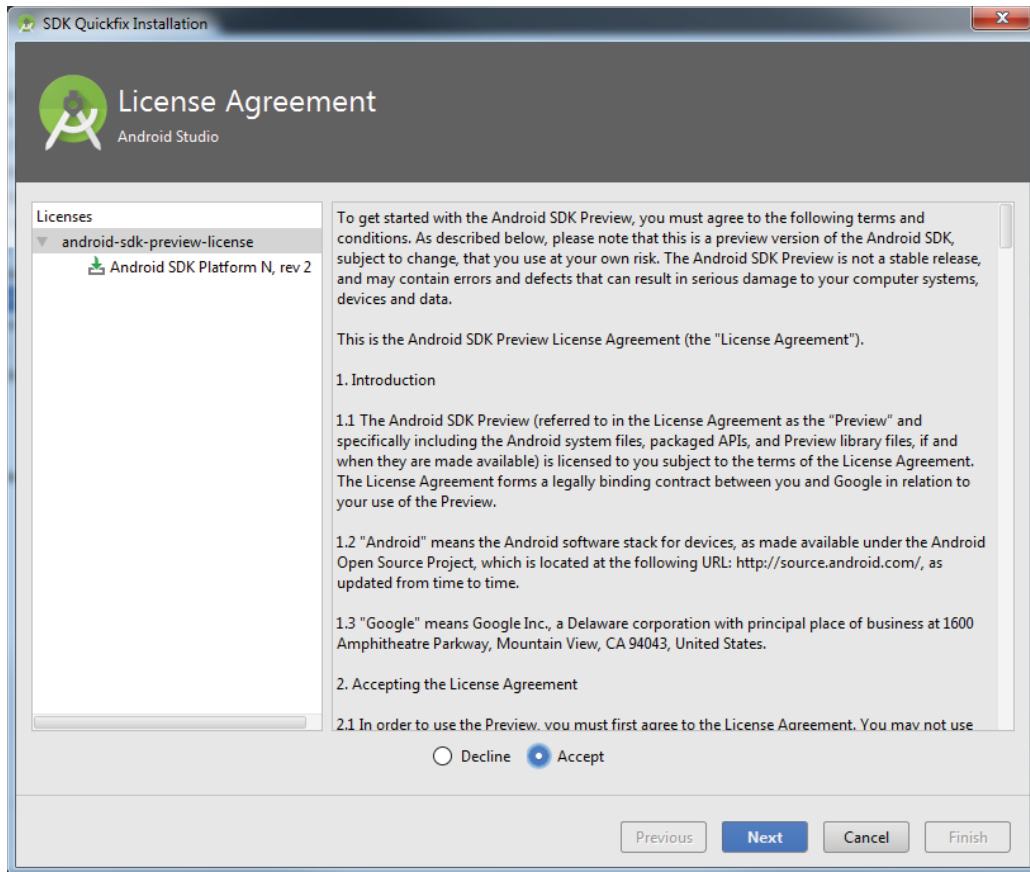


FIGURE 1-17

Use the following steps to create an AVD. This example demonstrates creating an AVD (put simply, an Android emulator) that emulates an Android device running Android N on the Nexus 5x hardware specs.

1. Start Android Studio so that the Welcome screen is visible (refer to Figure 1-15). Click Start a New Android Studio Project. You see the Create New Project Wizard shown in Figure 1-18.
2. Set up a HelloWorld project (that you will use in the final section of this chapter). Type **Chapter1HelloWorld** in the Application Name field.

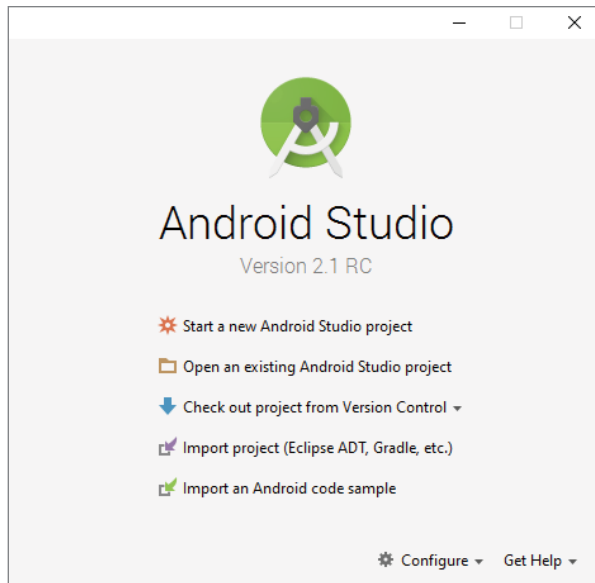


FIGURE 1-18

3. You can keep the default values for the other fields on the New Project screen (they will be explained in more detail in later chapters). Click Next.

NOTE For the purposes of setting up a quick Hello World project and creating an AVD, you will be accepting many of the default values, without explanation, during the project setup process. This is fine for now, as all of the settings are explained in much greater detail in subsequent chapters.

4. You should see the Targeted Android Devices screen. By default, the Create New Project Wizard selects for you the Android SDK level that has the greatest activity based on statistics gathered from Google Play. At the time this book was written 74.3 percent of the active devices on Google Play were written using Android Jelly Bean. For now, accept the default, as shown in Figure 1-19, and click Next.
5. On the Add an Activity to Mobile screen, accept the default choice—Empty Activity (see Figure 1-20)—and click Next.

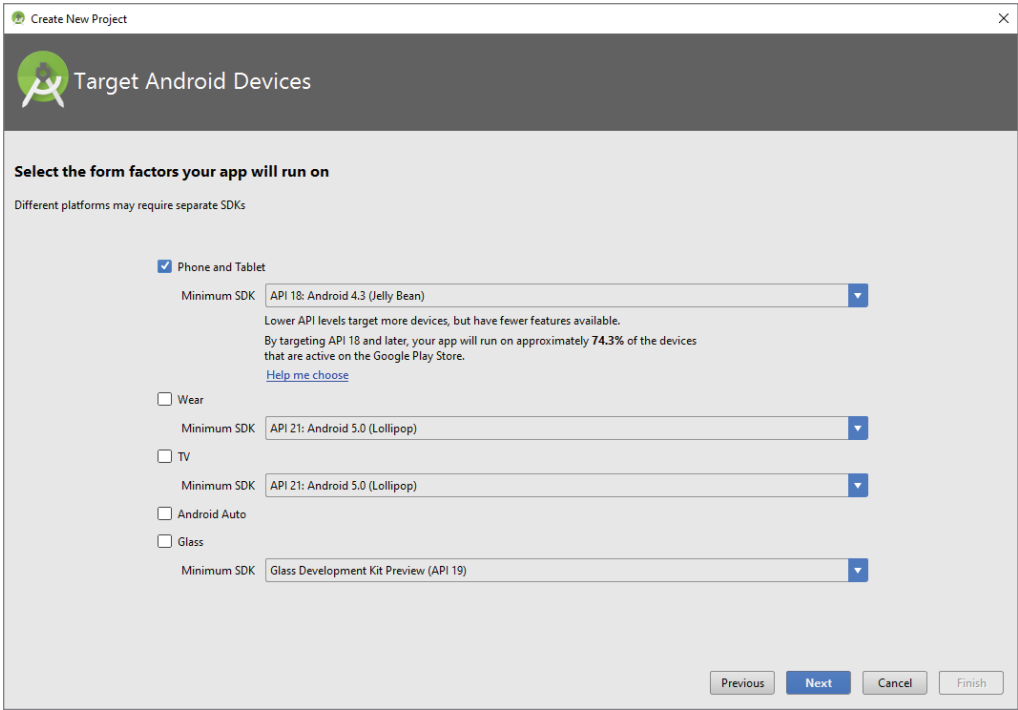


FIGURE 1-19

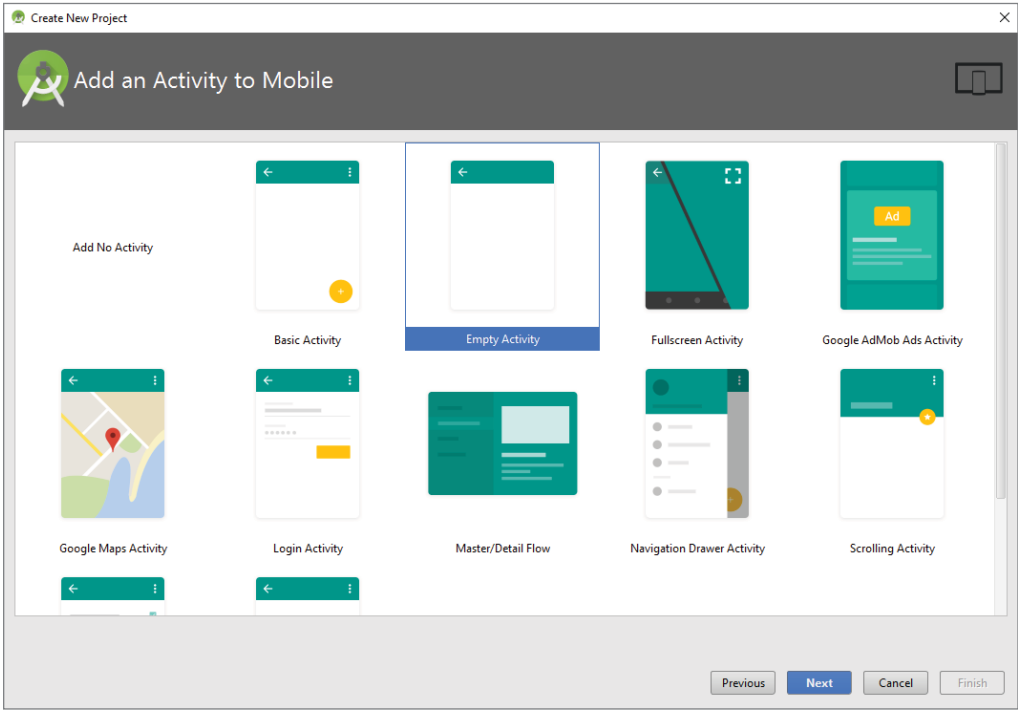


FIGURE 1-20

6. Accept all of the defaults on the Customize the Activity screen, as shown in Figure 1-21, and click Finish. Figure 1-22 shows the open Android Studio IDE.

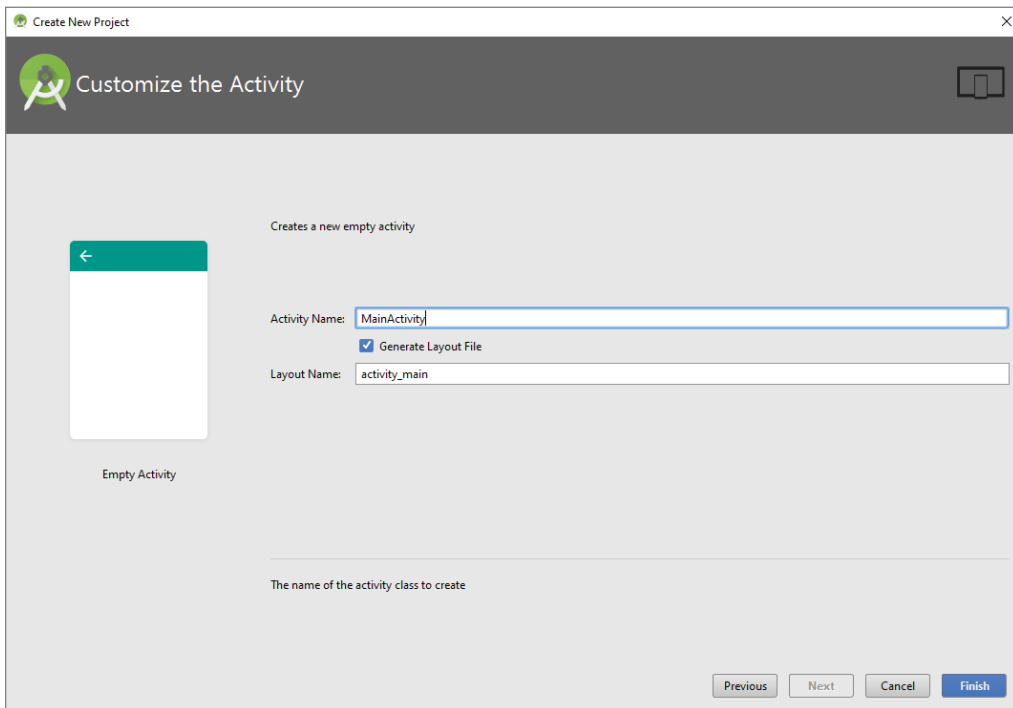


FIGURE 1-21

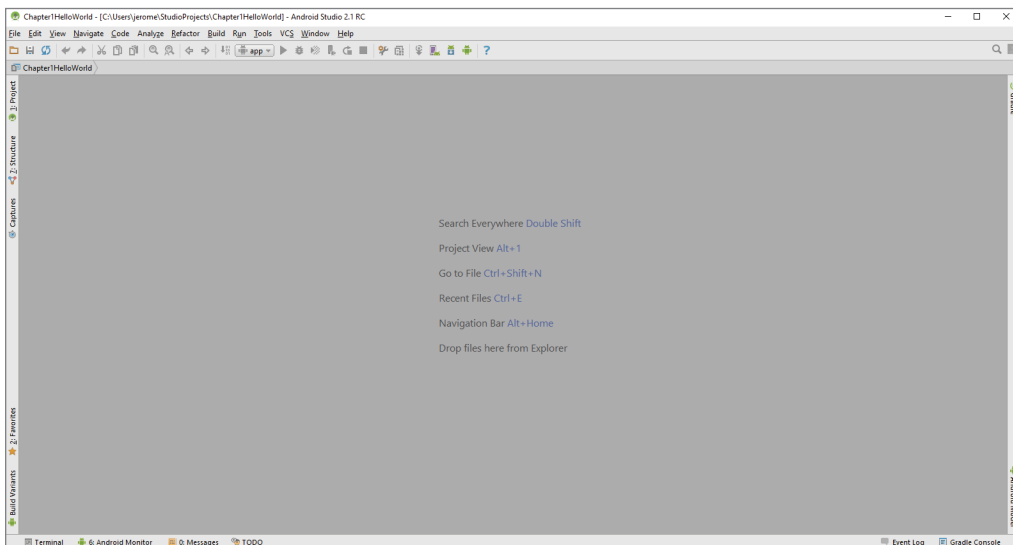


FIGURE 1-22

7. Launch the AVD Manager by selecting Tools ⇨ Android ⇨ AVD Manager or using the AVD Manager button from the toolbar. Figure 1-23 shows the Android Virtual Device Manager Wizard, which is where you set up AVDs to be used when you emulate your application in Android on your desktop.

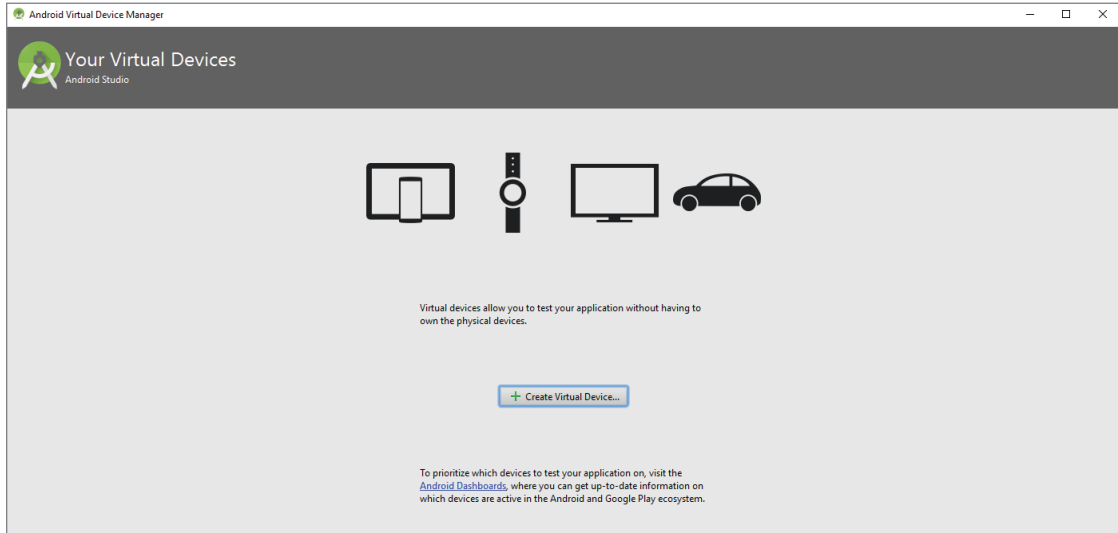


FIGURE 1-23

8. Click the + Create Virtual Device button to create a new AVD. The Virtual Device Configuration screen opens as shown in Figure 1-24.
9. Select the Nexus 5x hardware profile and click Next. Although none of the emulators offers the same performance as its actual hardware counterpart, the Nexus 5x should run well on most x86-based desktops, and it still offers some of the mid- to high-end Android device specs.
10. For the system image, select and install the latest option, which at the time this book was written is Android Nougat. Click the x86 Images tab (see Figure 1-25), select N from the list of images, and then click Next.

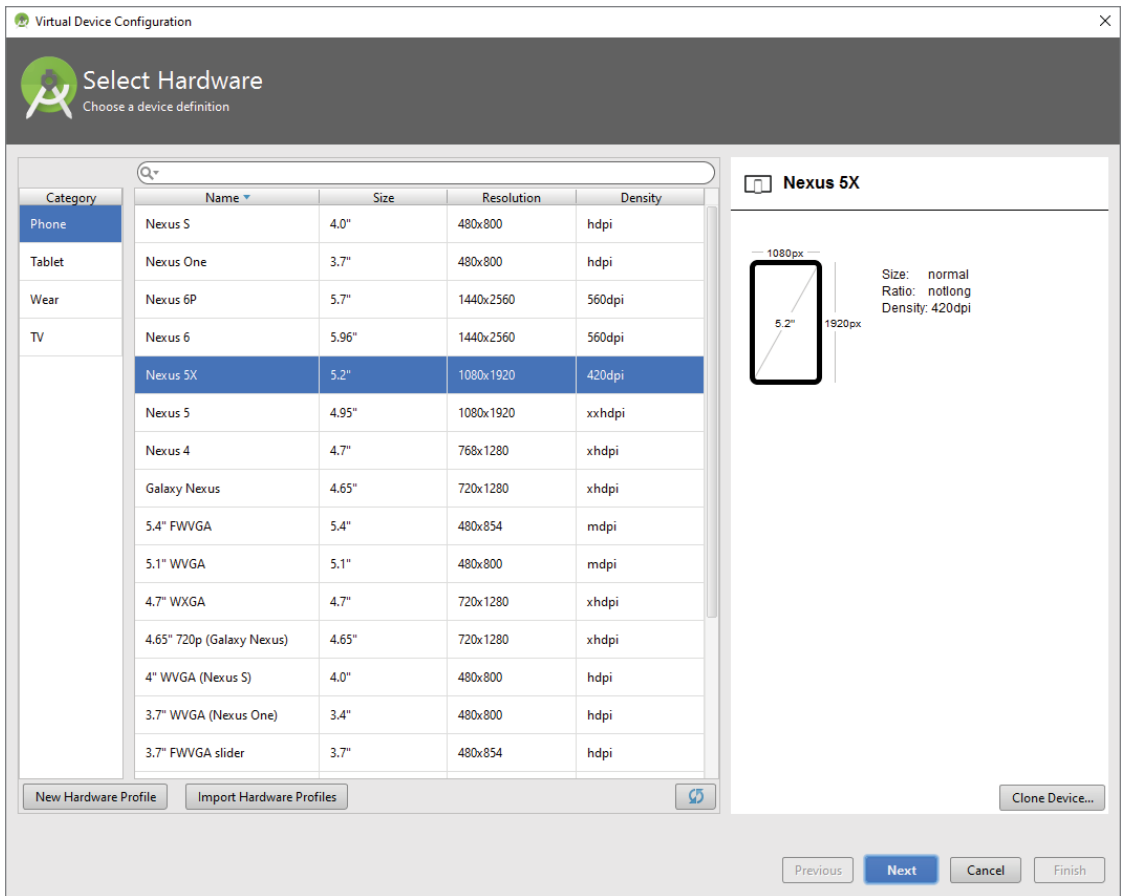


FIGURE 1-24

11. In the Android Virtual Device (AVD) dialog, accept the defaults as shown in Figure 1-26. Click the Finish button to begin building the AVD.

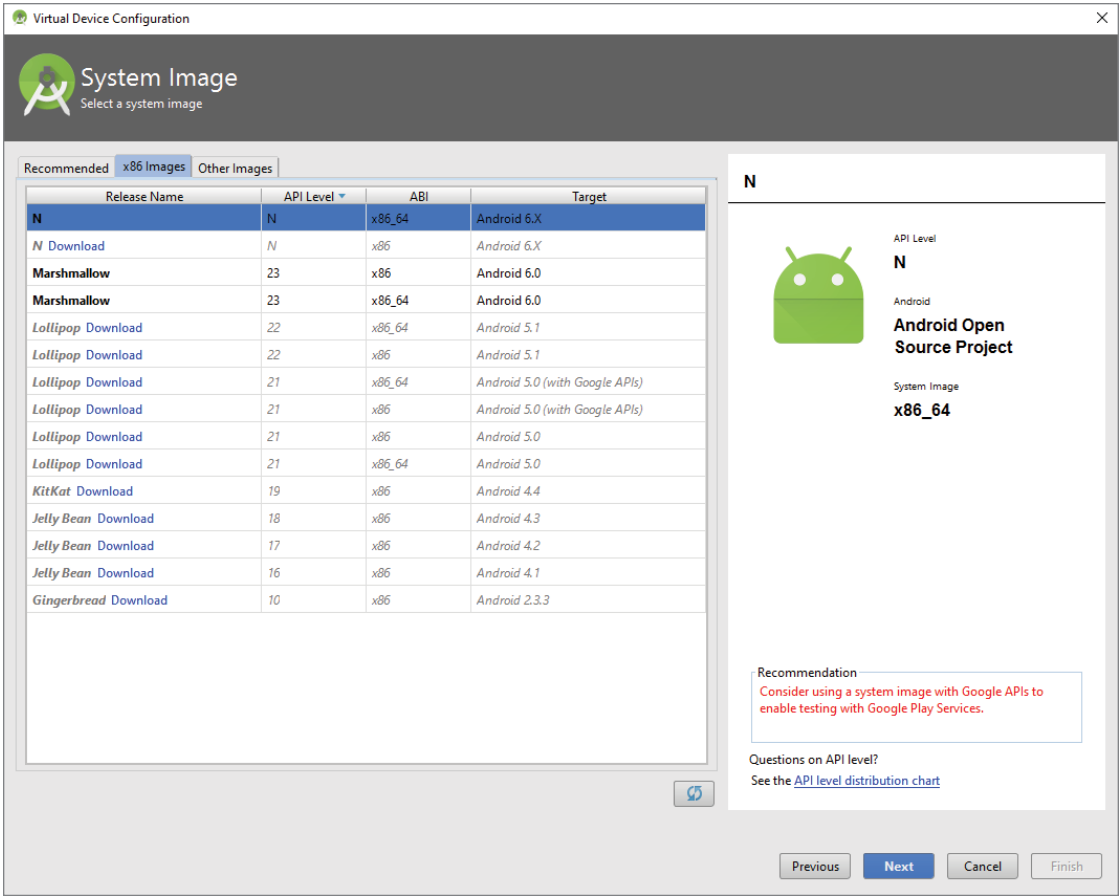


FIGURE 1-25

TIP It is preferable to create a few AVDs with different API levels and hardware configurations so that your application can be tested on different versions of the Android OS.