Get started with building your very own Android apps

Android Apps for Absolute Beginners

SECOND EDITION

Wallace Jackson

Apress®

Android Apps for Absolute Beginners

Second Edition

Wallace Jackson

Apress®

Android Apps for Absolute Beginners

ISBN 978-1-4302-4788-3

ISBN 978-1-4302-4789-0 (eBook)

President and Publisher: Paul Manning
Lead Editor: Steve Anglin
Developmental Editor: Douglas Pundick
Technical Reviewer: Chad Darby
Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Louise Corrigan, 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: Kevin Shea
Copy Editor: Judy Ann Levine
Compositor: SPi Global
Indexer: SPi Global
Artist: SPi Global
Cover Designer: Anna Ishchenko

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

For information on translations, please e-mail rights@apress.com, or visit 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.

Any source code or other supplementary materials referenced by the author in this text is available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code.

Contents at a Glance

Contents

About the Author

Wallace Jackson is the CEO of Mind Taffy Design, a new media content design and production company founded in 1991. Mind Taffy specializes in leveraging free for commercial use open source technologies to provide extremely compact data footprint, royalty-free, digital new media advertising and branding campaigns for leading international brands and manufacturers worldwide.

Wallace has been pushing the leading-edge of i3D and Rich Media Application Design via viral digital content deliverables, using under 1MB of Total Data Footprint, for over two decades. He has worked for leading international brands and manufacturers to create custom new media digital campaigns for industry-leading companies, including projects targeting brand marketing, PR, product demonstration, digital signage, e-learning, AdverGaming, product marketing, 3D logo design, and end-user training.

Wallace has produced cutting-edge new media projects in a number of digital media "verticals" or content deliverable areas, including: interactive 3D [i3D], Rich Internet Applications (RIA) content production, virtual world design, user interface (UI) design, user experience (UX) design, e-book design, multimedia production, 3D modeling, sound design, MIDI synthesis, music composition, image compositing, 3D animation, game programming, mobile application programming, BrandGame creation, website design, CSS programming, data optimization, digital imaging, digital painting, digital video editing, special effects, morphing, vector illustration, IPTV programming, iTV application design, interactive product demos, and tradeshow multimedia.

Wallace has created new media digital campaigns for leading international branded manufacturers, including Sony, Samsung, Tyco, Dell, Epson, IBM, Mitsubishi, Compaq, TEAC, KDS USA, CTX International, ADI Systems, Nokia, Micron, ViewSonic, OptiQuest, SGI, Western Digital, Sun Microsystems, ProView, Sceptre, KFC, ICM, EIZO, Nanao, Digital Equipment [DEC], TechMedia, Pacific Digital, ArtMedia, Maxcall, Altrasonic, DynaScan, EZC, Smile, KFC, Krillogy, and Kinoton GMBH.

Wallace holds an MSBA postgraduate degree in Marketing Strategy from the University of Southern California (USC), an MBA degree in Management Information Systems Design and Implementation from the USC Marshall School of Business, and a Bachelor's degree in Business Economics from the University of California at Los Angeles (UCLA) Anderson School of Management. He is currently the number 2 ranked All Time Top Expert on LinkedIn, out of more than 200,000,000 executives that currently use that business social media website. You can read Wallace's blog at `www.WallaceJackson.com` and follow him on Twitter @wallacejackson as well.

About the Technical Reviewer

Chád 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 his 15 years as a professional software architect, he has had the opportunity to work for Blue Cross/Blue Shield, Merck, Boeing, Northrop Grumman, 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 Sun Microsystems and IBM. He holds a BS in Computer Science from Carnegie Mellon University.

You can read Chád's blog at `www.luv2code.com` and follow him on Twitter @darbyluvs2code.

Acknowledgments

Steve Anglin, my lead editor, for his patience and thoughtful guidance in shaping this second edition of *Android Apps for Absolute Beginners*. Steve, thanks for guiding me as a returning Apress author, and I look forward to many future collaborations with you.

Chád Darby, my esteemed technical reviewer, for his hard work and insightful suggestions in shaping this second edition of this best-selling book.

Douglas Pundick, my Development Editor, for all his hard work helping me to write the best introductory Android book. I wouldn't have been able do it at all if it were not for you!

Kevin Shea, my Coordinating Editor, for listening to all of my miscellaneous and sundry problems during the writing of this book, and for helping to get them all sorted out.

Judy Ann Levine, my Copy Editor, for her excellent editing and book-polishing skills, and for all the great suggestions for making this a fantastic Android book.

Plus to the many loved ones and clients who patiently awaited my return to i3D content production from the "professional sidetracker" commonly known as writing an Android 4.1 programming book – I thank you for your patience…

—Wallace Jackson

Introduction

Over the last three years, Google's Android operating system (OS) has gone from a virtually unknown open source solution to the current mobile OS market leader among all mobile handsets, with over one-half of the market share and still climbing. Android has even started to dominate the tablet OS marketplace, and is also the foundation for the popular iTV OS known as GoogleTV as well as for e-book e-readers from Sony, Amazon (Kindle), and Barnes and Noble (Nook). There seems to be no end in sight for Android's rocketing success, which is great news for the owners of this book.

I've heard a great many people say, "I have a really phenomenal idea for a smartphone and tablet application! Can you program it for me!?" Rather than sit back and code all of these cool applications for everyone, I thought it might be a smarter idea to write a book about how an absolute beginner could code an Android application using open source tools that cost nothing to download and that are free for commercial use, and then leverage that new found knowledge to reach their dream of making their application idea a revenue-generating reality.

Thanks to open source tools and formats and Google's Android development environment, Oracle's Java programming language, Linus Torvalds' Linux operating system, the Eclipse code editing software, and to this book of course, vaporizing a software product out of thin air and at no production cost other than your PC and "sweat equity," is now a complete reality.

The Target: The Programming Neophyte

As you may have inferred from the title, this book assumes that you have never programmed before in any programming language. It is written for someone who has never written a single line of code, and who is thus unfamiliar with object-oriented programming (OOP) languages such as Oracle's Java and markup languages such as XML. Both of these open source languages are used extensively in creating Android applications and will be taught thoroughly in this book.

There are a lot of Java and Android books out there, but all of those books assume that you have programmed before and know all the OOP and programming lingo. I wanted to write a book that takes readers from knowing absolutely nothing about programming; not even knowing about how to install a software development kit (SDK) or an integrated development environment (IDE), all of the way from Ground Zero to being able to program useful Android applications using Java and XML and new media assets such as images, audio, and animation.

The Weapon: Android—An Innovative Internet 2.0 Coding Environment

Android is my Internet 2.0 development weapon of choice because it allows me to develop highly advanced applications for the primary Internet 2.0 devices, including the primary four consumer electronics product "verticals" where revenue potential is by far the greatest:

- Smartphones
- Tablets
- e-book e-readers
- iTVs or interactive television sets

The other reason I place my bets on Android is because it is open source and uses open source technologies and is therefore free from royalties and politics. It includes advanced new media "engines" (tools) such as OpenGL, ON2 VP8 (WebM and WebP), Java, XML, CSS, HTML5, PNG, and JPEG. I do not have to submit my Android application to any company and ask for permission to publish it, as long as it is not harmful in any way to others. For this reason, and due to the free for commercial use nature of open source software, there is little external risk involved in developing an application for the Android Platform.

How This Book Is Organized

Because this is a book for absolute beginners, we start at the very beginning—showing where to download, and how to install, the various Android, Java, and Eclipse environments. We also show how to configure these environments, and how to set them up for application development and testing. We even show how and where to download the other leading-edge new media tools (GIMP, for instance) that you will use in conjunction with the primary Android development tools.

We essentially show you exactly how to put together a complete and professional-level Android New Media Content Production Workstation, and at zero cost to yourself to boot. This in itself is no easy task, and must be done correctly, as these professional tools provide the foundation for all of our Android development, debugging, and testing for the remainder of the book.

Next we provide you with an overview of where Android came from, why, how, and when Google acquired it, and how it is uniquely structured among software development platforms. We introduce XML, Java, OOP, and Android concepts soon after that, as well as covering how Android manages its screen layout. We then move these concepts into use in later chapters in the second half of the book; these chapters explain the most important concepts in Android, in their most logical order, as they pertain to your applications development.

In that second half of the book, we start getting into developing a user interface (UI), as that is the front-end or interface for your end-users to your Android application. Soon after that we cover how your UI talks to your application via events processing. To spice up your application's visual appearance, we'll get into graphics, animation, and audio, and then into even more advanced topics after that, such as databases and communications.

Finally, we will look at some of the advanced features of Android that you will want to visit after finishing the book; these are topics that are too advanced for a first book on Android, but which provide some of the coolest features to be found in smartphone, iTV, and tablet software development today.

We'll walk you through all of these topics and concepts with screenshots of the IDE and visual examples, and then take you through step-by-step examples reinforcing these concepts. Sometimes we may even repeat previous topics to reinforce what you have learned and apply these programming skills in new and different ways. This enables new programmers to reapply key development skills and feel a sense of accomplishment as they progress.

The Formula for Success

Learning to develop an Android application is an interactive process between you and the tools and technologies (Eclipse, XML, Java, Android, GIMP, and so on) that I cover in this book. Just like learning to play a sport, you have to develop these skills and practice them daily. You need to work through the examples and exercises in this book, more than once if necessary, to become comfortable with each concept and proficient in their execution.

Just because you understand a concept, that doesn't necessarily mean you will know how to apply it creatively and use it effectively. That takes practice and it ultimately will happen when the "ah-ha" moment occurs, when you understand a concept in context with the other concepts that interconnect with it.

You will learn quite a bit about how Android works from this introductory book. You will glean a lot of insight into the inner working of Android by working through all of the exercises in this book. But you will also learn new things not specifically mentioned in this book when you compile, run, and debug your programs. Spending time experimenting with your code and trying to find out why it is not working the way you want, or trying to add new features, is a learning process that is also very valuable.

The downside of debugging is it can sometimes be quite frustrating to the new developer. If you have never wanted to put a bullet in your computer display, you will soon. You will question why you are doing this, and whether you are savvy enough to solve the problem. Programming can be very humbling, even for the most experienced of applications developers.

As with an athlete, the more you practice, the better you will become at your skill. You can do some truly amazing things as an Android programmer. The world is your oyster. It is one of the most satisfying accomplishments you can have, seeing your app in the Google Play (Android App) Store. However, there is a price, and that price is time spent practicing your coding.

Here is our formula for success:

- Trust that you can pull it off. You may be the only one who says you can't do this. Don't tell yourself that.
- Work through all the examples and exercises in this book, twice if necessary, until you understand them.
- Code, code some more, and keep coding—don't stop. The more you code, the better you'll get.

- Do further research via Google Search as well as the Developer.Android.com website into areas of the Android OS that interest you, and that you want to master.
- Be patient with yourself. If you were fortunate enough to have been a star pupil who could memorize material simply by reading it, this will probably not happen with Java and XML coding. You are going to have to spend a lot of time coding to come to understand what exactly is happening inside the Android OS Environment.
- Whatever you do: DON'T GIVE UP!

Required Software, Materials, and Equipment

One of the great things about Java, Android, and Eclipse is they are available on all three primary operating systems that are in use today:

- Windows
- Macintosh
- Linux

The other great thing about Java, Android, and Eclipse is that they are completely free. For equipment, any modern dual-core or quad-core computer will do. Fortunately computer workstations are only $200 to $400 brand new on `www.PriceWatch.com` or you can walk into WalMart and buy an HP or Acer tower with a fast quad-core processor for $300 to $500 including Windows 7 or 8. There are also open source OSes such as SUSE Linux, which is free and an amazing development operating system. SUSE Linux V12 can be downloaded at `www.OpenSUSE.com` and is currently at version 12.2 and is very stable.

Operating System and IDE

Although you can use Android on many platforms, the Eclipse integrated development environment (IDE) that developers use to develop Android apps is most commonly used on an Intel-based Windows or Linux PC. The Eclipse Juno 4.2 for Java EE IDE is free, and is available on the Internet at `www.eclipse.org`. The operating system should be Windows XP SP3 or later, or SUSE Linux 12.2 or later, to run Eclipse most effectively. Note that as of Android 4.2 API Level 17, developers can now develop using a 64-bit "clean" Android IDE environment, so the Windows 7 64-bit or Windows 8 64-bit OSs may be the best way to go if you want to develop using a 64-bit platform.

Software Development Kits

You will need to download the Eclipse Juno 4.2 for Java EE IDE from Eclipse.org and the Android 4.1 or later SDK from Google. This is available at `http://developer.android.com/SDK/`. This is another area that changed significantly with the release of Android 4.2, as there is now an ADT Bundle that can be downloaded for either 32-bit or 64-bit OSs which makes installation much easier than it used to be. We cover this in Chapter 3, and do the install in both ways, so that you can see the long-form installation (and see how everything goes together in the process) as well as the streamlined installation that

emerged along with the Android 4.2 Level 17 API in the end of 2012. Also note that wherever we reference Android 4.1 or Android 4.1.2 you can now substitute Android 4.2, as we have updated several chapters with the new 4.2 features. Whew! Caught that one just in time!

Dual Displays

It is highly recommended that developers have a second display connected to their computer. It is great to step through your code and watch your output window and Android emulator at the same time on dual, independent displays. Today's PC hardware makes this easy. Just plug your second display in to the second display port of any Intel-based PC or laptop, with the correct display port adapter, of course, and you're able to have two displays working independently from one another. Note it is not required to have dual displays. You will just have to organize your open windows to fit on your screen if you don't. I am using a Philips 32" HDTV 1920 by 1080 LCD display to code on, so that my code is very readable. With 40" HDTV displays at $250 at WalMart, having a big widescreen or two to use for your Android application development workstation is a great idea!

Chapter 1

Preliminary Information: Before We Get Started

This chapter introduces the Android operating system, giving you a little background information to help put things into perspective. We'll visit just how expansive this platform has become in today's Internet 2.0 environment of portable consumer electronic devices. *Internet 2.0* here refers to the consumption of the Internet over a wide variety of different types of data networks using highly portable consumer electronic devices, including smartphones, tablets, e-book readers, and even new emerging consumer electronic products such as interactive television (iTV).

As this is an introductory book on the subject not all of the advanced new media-related areas, such as 3D and video streaming, will be covered. Some specifics of what the book will and will not cover are outlined in this chapter.

At the end of the chapter, you'll learn which tools you will need to obtain to develop for the Google Android platform, with instructions on how to download them.

Those of you who already recognize the significance of the Android revolution and know which tools are needed to develop Android applications development may want to skip this chapter. However, there may be some tidbits in here that could spawn development ideas—so skip over it at your own risk!

Just a bit of fair warning: developing reliable applications for Android is not in any way a trivial task. It takes a fair amount of knowledge of both high-level programming languages such as Java and markup languages such as XML. Building useful and engaging new media applications also requires a deep knowledge of related new media technologies such as 2D imaging, 3D rendering, audio processing, video streaming, GPS localization, and database design.

Don't expect to learn all of this at one sitting. Becoming a top-notch Android programmer will take years of dedication and practice as well as diligent research and trial and error. In this book, you will gain the foundation that you need to build future expertise as well as learn the work process for eventually building your own Android masterpiece.

Some History: What Is Android?

Android was originally created by Andy Rubin as an operating system for mobile phones, around the dawn of this twenty-first century. In 2005, Google acquired Android Inc., and made Andy Rubin the Director of Mobile Platforms for Google. Many think the acquisition was largely in response to the emergence of the Apple iPhone around that time; however, there were enough other large players, such as RIM Blackberry, Nokia Symbian, and Microsoft Windows Mobile, that it seemed a salient business decision for Google to purchase the talent and intellectual property necessary to assert the company into this emerging space, which has become known as Internet 2.0.

Internet 2.0 allows users of consumer electronics to access content via widely varied data networks through highly portable consumer electronic devices, such as smartphones, touchscreen tablets, and e-book e-Readers, and even through not so portable devices, such as iTVs, home media centers, and set-top boxes. This puts new media content such as games, 3D animation, digital video, digital audio, and high-definition imagery into our lives at every turn. Android is one of the vehicles that digital artists will increasingly leverage to develop new media creations that users have never before experienced.

Over the past decade, Android has matured and evolved into an extremely reliable, bulletproof, embedded operating system platform, having gone from version 1.0 to stable versions at 1.5, 1.6, 2.0, 2.1, 2.2, 2.3, 3.0, 3.1, 3.2, 3.3, 4.0, and recently, 4.1. Here are the latest stats from the Android website at: `http://developer.android.com/about/dashboards/index.html`

VERSION	CODENAME	API LEVEL	MARKET SHARE
1.5	Cupcake	3	0.2 %
1.6	Donut	4	0.4 %
2.1	Eclair	7	3.7 %
2.2	Froyo	8	14 %
2.3.2	Gingerbread	9	0.3 %
2.3.7	Gingerbread	10	57.2 %
3.1	Honeycomb	12	0.5 %
3.2	Honeycomb	13	1.6 %
4.0.2	Ice Cream Sandwich	14	0.1 %
4.0.4	Ice Cream Sandwich	15	20.8 %
4.1	Jelly Bean	16	1.2 %

An embedded operating system is like having an entire computer on a chip small enough to fit into handheld consumer electronics, but powerful enough to run applications (commonly known as *apps*). Like today's computers, Internet 2.0 devices such as smartphones, tablets, e-readers, and iTVs now feature dual-core and even quad-core computer processing power as well as one or two gigabytes of system memory.

Android has the power of a full-blown computer operating system. It is based on the Linux open source platform and Oracle's (formerly Sun Microsystems's) Java, one of the world's most popular programming languages.

> **Note** The term *open source* refers to software that has often been developed collaboratively by an open community of individuals, is freely available for commercial use, and comes with all of its source code so that it can be further modified if necessary. Android is open source, though Google develops it internally before releasing the source code; from that point on, it is freely available for commercial use.

It is not uncommon for an Android product to have a 2GHz processor and 2GB of fast, computer-grade DDR2 memory. This rivals desktop computers of just a few years ago and `netbooks` that are still currently available. You will see a further convergence of handheld operating systems and desktop operating systems as time goes on. Some examples are the new Windows 8 operating system and Linux platform.

Once it became evident that Android and open source were forces to be reckoned with, nearly 100 major companies—including HTC, Samsung, LG Electronics, and T-Mobile—formed and joined the Open Handset Alliance (OHA). This was done to put some momentum behind Google's open source Android platform, and it worked. Today, more brand manufacturers use Android as an operating system on their consumer electronic devices than any other operating system.

The development of the OHA is a major benefit to Android developers. Android allows developers to create their applications in a single environment, and the support by the OHA allows developers to deliver their content across dozens of major branded manufacturer's products, as well as across several different types of consumer electronic devices: smartphones, iTV sets, e-book e-readers, home media centers, set-top boxes, and touchscreen tablets. Exciting possibilities—to say the least.

So, Android is a seasoned operating system that has become one of the biggest players in computing today, and with Google behind it. Android uses freely available open source technologies such as Linux and Java, and standards such as XML, to provide a content and application delivery platform to developers as well as the world's largest consumer electronics manufacturers. Can you spell O-P-P-O-R-T-U-N-I-T-Y? I sure can … it's spelled *ANDROID*.

Advantage Android: How Can Android Benefit Me?

There are simply too many benefits of the Android platform to ignore Android development.

First of all, Android is based on open source technology, which was at its inception not as refined as paid technologies from Apple and Microsoft. However, over the past two decades, open source software technology has become equally as sophisticated as conventional development technologies. This is evident in Internet 2.0, as the majority of the consumer electronics manufacturers have chosen Linux and Java over the Windows and Macintosh operating systems. Therefore, Android developers can develop not only for smartphones, but also for new and emerging consumer electronic devices such as tablets and iTVs that are network compatible and thus available to connect to the Android Marketplace, recently rebranded by Google as Google Play. This translates into more sales onto more devices in more areas of the customer's life, and thus more incentive to develop for Android over closed technologies such as Windows and iOS and over less popular and less prolific PC operating systems.

In addition to being free for commercial use, Android has one of the largest, wealthiest, and most innovative companies in modern-day computing behind it: Google. Add in the OHA, and you have more than a trillion dollars of megabrand companies behind you supporting your development efforts. It seems too good to be true, but it's a fact, if you are an Android developer (which you are about to be, in about a dozen chapters) then you've got a great support team behind you!

Finally, and most important, it's much easier to get your Android applications published than those for other platforms that are similar to Android (I won't mention any names here to protect the not so innocent). We've all heard the horror stories regarding major development companies waiting months, and sometimes years, for their apps to be approved for the app marketplace. These problems are nearly nonexistent on the open source Android platform. Publishing your app on the Google Play Android marketplace is as easy as paying $25, uploading your *.apk* file, and specifying free or paid download.

The Scope of This Book

This book is an introduction to developing applications for Android. It's intended for absolute beginners—that is, people who have never created an application on the Android platform for a consumer electronic device. I do not assume that you know what Java is or how XML works.

What's Covered

This book covers the basic and essential elements of Android development, including the following

- The open source tools required to develop for this platform.
  - Where to get these free tools.
  - How to properly install and configure the necessary tools for application development.
  - Which third-party tools are useful to use in conjunction with the Android development tools.
  - Which operating systems and platforms currently support development for the Android using these tools.
- The concepts and programming constructs for Java and XML, and their practical applications in creating Android applications.
- How Android goes about setting up an Android application.
  - How it defines the user interfaces.
  - How it writes to the display screen.
  - How it communicates with other Android applications.
  - How it interfaces with data, resources, networks, and the Internet.
  - How it alerts users to events that are taking place inside and outside the application.

- How Android applications are published.
- How Android applications are ultimately sold, downloaded, and updated automatically through the Google Play Android marketplace.

You should realize that Android has more than 44 Java packages that contain over 7,000 pieces of programming code functionality to allow you to do just about anything imaginable—from putting a button on the screen to synthesizing speech and accessing advanced smartphone features like the high-resolution camera, GPS, and accelerometer.

> **Note** A *package* in Java is a collection of programming utilities that all have related and interconnected functionality. For example, the java.io package contains utilities to deal with input and output to your program, such as reading the contents of a file or saving data to a file. Later chapters describe how to organize your own code into packages.

What does this mean? It means that even the most advanced Android books cannot cover the plethora of things that the Android platform can do. In fact, most books specialize in a specific area in the Android APIs. There is plenty of complexity in each API, which ultimately, from the developer's viewpoint, translates into incredible creative power. "What's the price?" you might ask. Your time spent in mastering each API is the only price you will pay, as Android is otherwise free for commercial use.

What's Not Covered

So, what *isn't* covered in this book? What cool, powerful capabilities do you have to look forward to in that next level book on Android programming?

On the hardware side, we will not be looking at how to control the camera, access GPS data from the smartphone, and access the accelerometer and gyroscope that allow the user to turn the phone around and have the application react to phone positioning. We will not be delving into advanced touchscreen concepts such as gestures, or accessing other hardware such as the microphone, Bluetooth, and wireless connections.

On the software side, we will not be diving into creating your own Android SQLite database structure, or its new media codecs for digital video and digital audio, and its real-time 3D rendering system (called OpenGL ES 2.0). We will not be exploring speech synthesis and recognition, or the universal language support that allows developers to create applications that display characters correctly in dozens of international languages and foreign character sets. We will not be getting into advanced programming such as game development, artificial intelligence, and physics simulations. All of these topics are better suited to books that focus on these complex and detailed topical areas.

Preparing for Liftoff: SDK Tools to Download

In Chapter 3, you'll learn how to set up a complete Android development environment. We focus on Windows 7 because that's what the vast majority of developers use to develop for Android, but the process on Mac and Linux systems is similar, and I'll make sure you can follow along if you

prefer either of those systems. Also, because the Android r20.0.3 SDK, known as the Android Jelly Bean 4.1 environment, uses a 32-bit programming environment, we will be using the latest Java 6 update 37 in its 32-bit version, and the Eclipse 4.2.1 Juno for Java EE 32-bit integrated development environment (IDE) software, both of which work perfectly on 32-bit Windows OSs (such as XP, Vista, and Win7) as well as on the 64-bit Windows 7 and Windows 8 OSs.

Here, we'll look at where to go to download the tools you'll need, so that you are ready for action when the time comes to install and configure them. This is because each of these development tools is hundreds of megabytes in file size, and depending on your connection speed, may take anywhere from ten minutes to ten hours to download.

There are three major components of an Android development environment:

- Java
- Eclipse
- Android

In Chapter 3, when you install and configure the packages you are downloading now, you will see that Eclipse requires the Java package to be installed in order to install and run. Therefore, we will walk through downloading them in the order of installation, from Java to Eclipse to Android.

Java

Let's start with the foundation for everything we are doing, the Java 6 Platform, Standard Edition (Java SE) 32-bit version. Java 6 SE contains the core Java programming language. Note that Android 4.1 does not yet support the use of Java 7.

To download Java 6 SE, simply go to the Java SE Downloads section of Oracle's web site, which is in the Technology Network section under the Java directory, at this URL:

`http://www.oracle.com/technetwork/java/javase/downloads/index.html`

Figure 1-1 shows the Java SE Downloads site. Be sure and download Java 6 and not Java 7.

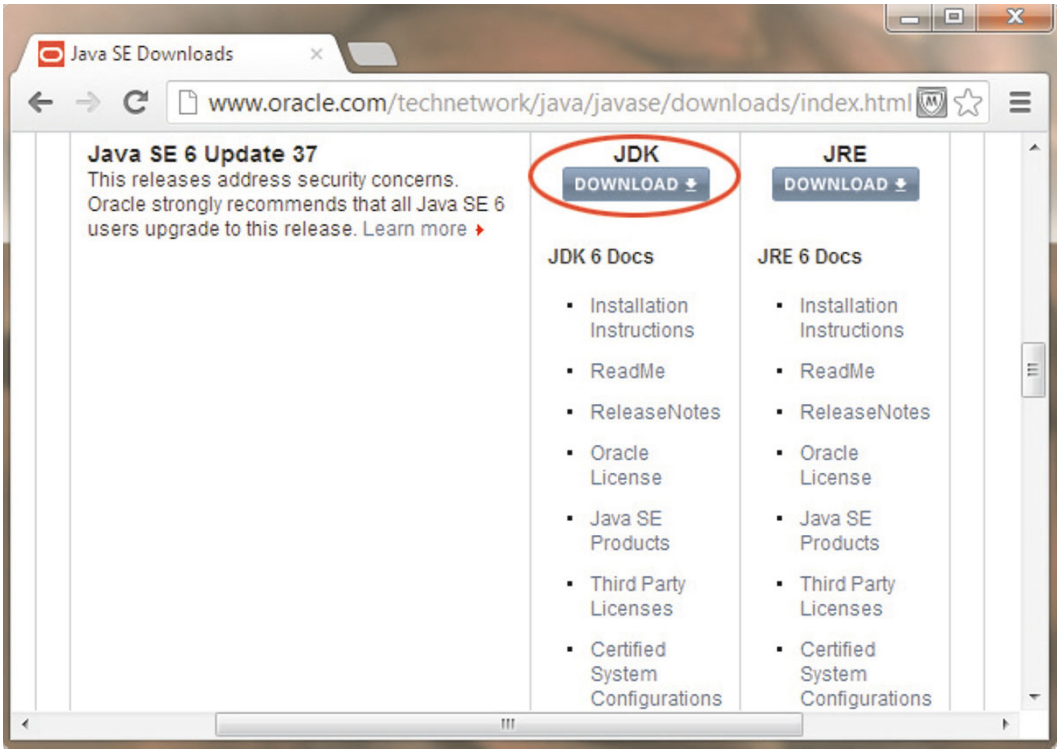

Figure 1-1. *Download the Java SE 6 JDK*

Click the Download Java 6 JDK button to start downloading the Java 6 SE Java Development Kit (JDK). This will take you to a second page shown in Figure 1-2 where you need to Accept the License Agreement by selecting the shown radio button option and then download the Windows 32-bit version of the Java 6 software also highlighted in the screenshot below by clicking on the link shown in red.

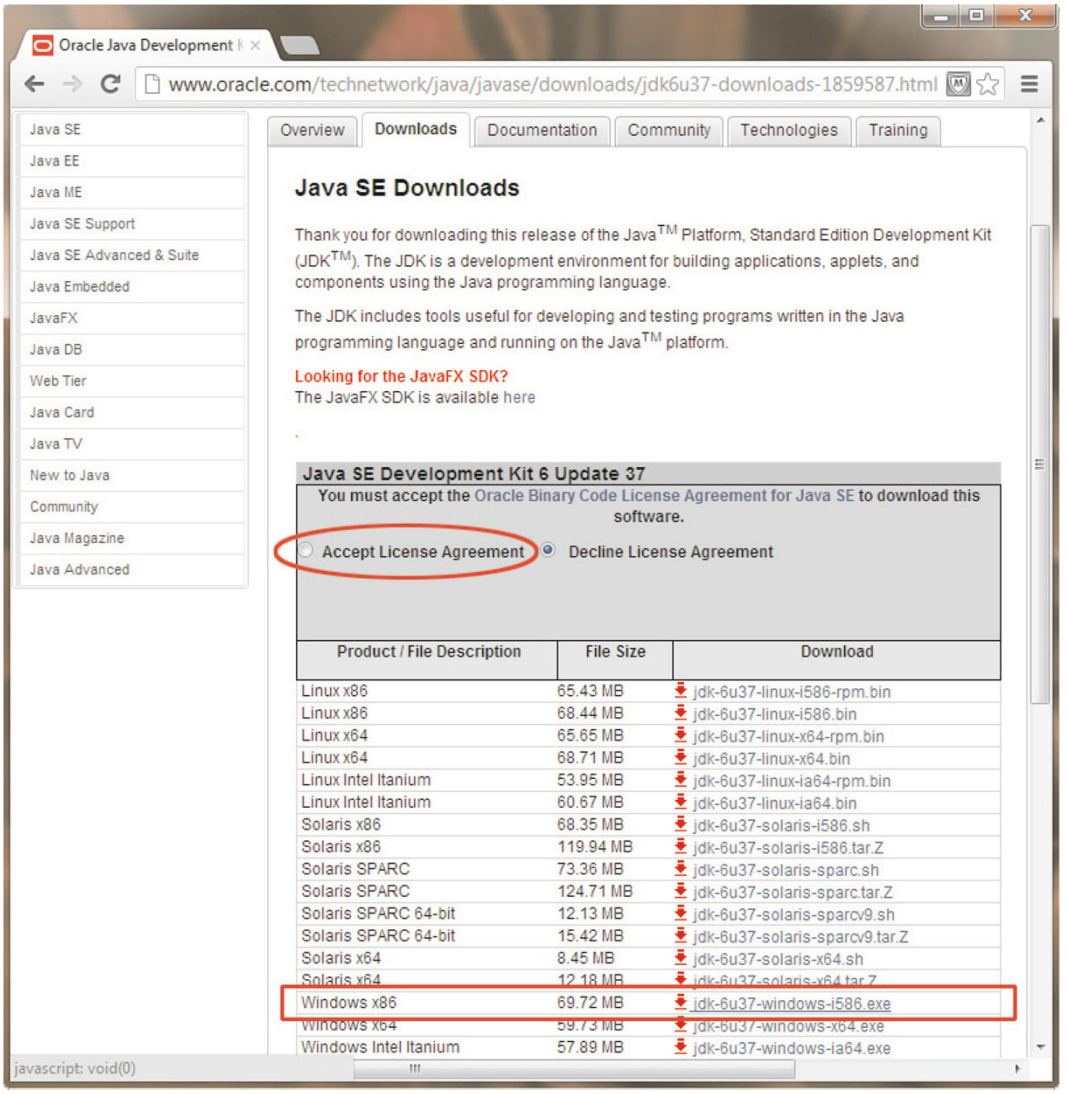

Figure 1-2. Accept License Agreement and Download Windows x86 version of Java 6

Note Make sure *not* to download Java 7 Platform, Enterprise Edition (Java EE), JavaFX 2.2, or Java with NetBeans. These are the buttons on the top of the first download page. Scroll down to the bottom and find Java 6 (shown in Figure 1-1).

Eclipse

Eclipse is an *integrated development environment* (IDE), which is a piece of software dedicated to allowing you to easily write programming code and run and test that code in an integrated environment. In other words, you write all your code into its text editor, before running and testing that code using commands in Eclipse, without ever needing to switch to another program.

Currently, Android requires the Eclipse IDE, and I recommend the Eclipse Juno Version 4.2 for Java EE. You should download a version of Eclipse that supports Java—such as the Eclipse Juno 4.2 IDE for Java EE shown in Figure 1-3. Go to the Eclipse website Downloads section at this URL:

`http://www.eclipse.org/downloads/`

Figure 1-3 shows the Eclipse Juno 4.2 for Java EE software package that you should download.

Click the Windows 32-bit version link in the right-hand column, and your download begins.

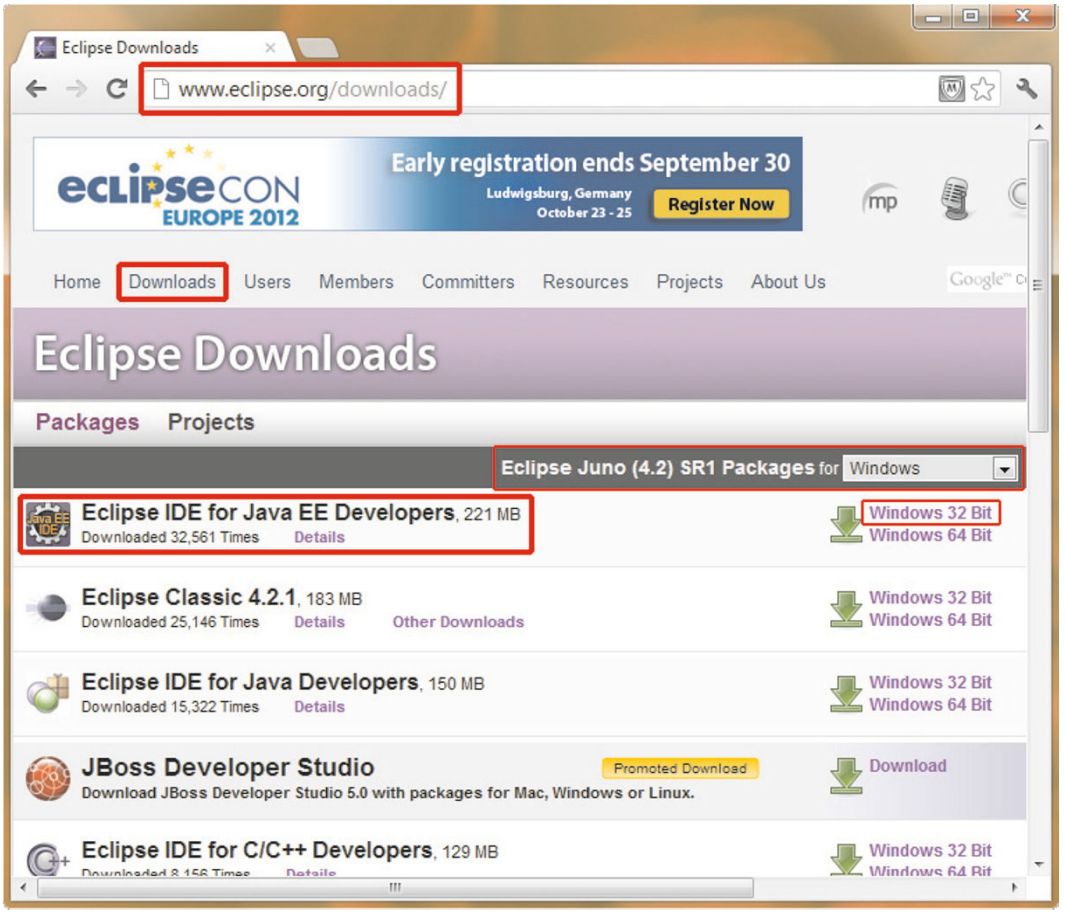

Figure 1-3. *Choose to download the Eclipse 4.2.1 Juno for Java EE IDE for Java Developers*