Android Studio IDE Quick Reference

A Pocket Guide to Android Studio Development

—

Ted Hagos

Apress®

Android Studio IDE Quick Reference

A Pocket Guide to Android Studio Development

Ted Hagos

Apress®

Android Studio IDE Quick Reference: A Pocket Guide to Android Studio Development

Ted Hagos
Manila, National Capital Region, Philippines

ISBN-13 (pbk): 978-1-4842-4952-9 ISBN-13 (electronic): 978-1-4842-4953-6
https://doi.org/10.1007/978-1-4842-4953-6

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 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.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Steve Anglin
Development Editor: Matthew Moodie
Coordinating Editor: Mark Powers

Cover designed by eStudioCalamar

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. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail editorial@apress.com; for reprint, paperback, or audio rights, please email bookpermissions@springernature.com.

Apress titles 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 Print and eBook Bulk Sales web page at www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484249529. For more detailed information, please visit www.apress.com/source-code.

Printed on acid-free paper

For Adrianne and Stephanie.

Contents

About the Author

Ted Hagos is the CTO and Data Protection Officer of RenditionDigital International, a software development company based out of Dublin. Before he joined RDI, he had various software development roles and also spent time as trainer at IBM Advanced Career Education, Ateneo ITI, and Asia Pacific College. He spent many years in software development dating back to the days of Turbo C, Clipper, dBase IV, and Visual Basic. Eventually, he found Java and spent many years working with it. Nowadays, he's busy with full-stack JavaScript and Android.

About the Technical Reviewers

Marcos Placona is a developer evangelist at Twilio and a GDE. He serves communities in London and all over Europe. He is passionate about technology and security and he spends a great deal of his time building mobile and web apps and occasionally connecting them to physical devices. Marcos is a great believer in open source projects. When he's not writing open source code, he's probably blogging about code at https://androidsecurity.info, https://androidthings.rocks, or https://realkotlin.com. He's also a great API enthusiast and believes they bring peace to the software engineering world.

Massimo Nardone has more than 24 years of experience in security, web/mobile development, cloud, and IT architecture. His true IT passions are security and Android. He has been programming and teaching how to program with Android, Perl, PHP, Java, VB, Python, C/C++, and MySQL for more than 20 years. He holds a Master of Science degree in Computing Science from the University of Salerno, Italy. He has worked as a Project Manager, Software Engineer, Research Engineer, Chief Security Architect, Information Security Manager, PCI/SCADA Auditor, and Senior Lead IT Security/Cloud/SCADA Architect. His technical skills include security, Android, cloud, Java, MySQL, Drupal, Cobol, Perl, web and mobile development, MongoDB, D3, Joomla, Couchbase, C/C++, WebGL, Python, Pro Rails, Django CMS, Jekyll, Scratch, etc. He was a visiting lecturer and supervisor for exercises at the Networking Laboratory of the Helsinki University of Technology (Aalto University). He holds four international patents (PKI, SIP, SAML, and Proxy areas). He currently works as the Chief

Information Security Officer (CISO) for Cargotec Oyj and he is member of the ISACA Finland Chapter Board. Massimo has reviewed more than 45 IT books for different publishers; he is the coauthor of *Pro JPA in Java EE 8* (Apress, 2018), *Beginning EJB in Java EE 8* (Apress, 2018), and *Pro Android Games* (Apress, 2015).

Acknowledgments

To Stephanie and Adrianne, my thanks and my love.

Introduction

Welcome to *Android Studio IDE Quick Reference*. I wrote this book to serve as a handy reference to the notable capabilities of Android Studio.

This book is comprised of 13 short chapters. Each chapter breezes through some features of Android Studio. While the book is focused on the IDE, many chapters actually deal with Android programming as well, so you're going to see some code samples.

Who This Book Is For

This book is for the experienced programmer who needs a quick reference on how to do some common tasks in Android Studio IDE. The code examples are in Java, so it's for those devs who build Android apps in Java. This book might also work for someone new to Android Studio but not new in Android development—such as an Eclipse user who wants to try out Android Studio.

Source Code

Source code for this book can be downloaded by clicking the Download Source Code button located at www.apress.com/us/book/9781484249529.

Chapter 1

Setup

What this chapter covers:

- Installing Android Studio
- Setting up the IDE
- Basic configuration

Developing applications for Android was not always as convenient as it is today. When Android 1.0 was released in 2008, what developers got by way of a development kit was little more than a handful of command line tools and Ant build scripts. Building apps with Vim, Ant, and other command line tools wasn't so bad if you're used to that kind of thing, but many developers were not used to it. The lack of IDE capabilities like code hinting, project setups, and integrated debugging was somewhat of a barrier to entry.

Thankfully, Android Development Tools (ADT) for the Eclipse IDE were released, also in 2008. Eclipse was, and still is, a favorite and the IDE of choice for many Java developers. It was natural that it would also be the go-to IDE for Android developers.

From 2009 to 2012, Eclipse remained the preferred IDE for Android development. The Android SDK has undergone both major and incremental changes in structure and in scope. In 2009, the SDK manager was released; it was used to download tools, individual SDK versions, and Android images for use with the emulator. In 2010, additional images were released for the ARM processor and x86 CPUs.

2012 was a big year because Eclipse and the ADT were finally bundled. Until that time, developers had to install Eclipse and ADT separately, and the installation process wasn't always smooth. So the bundling

T. Hagos, *Android Studio IDE Quick Reference*,
https://doi.org/10.1007/978-1-4842-4953-6_1

of the two made it a whole lot easier to get started with Android development. 2012 also marked the last year of Eclipse being the dominant IDE for Android.

In 2013, Android Studio (AS) was released. To be sure, it was still beta, but the writing on the wall was clear: it would be the official IDE for Android development. Android Studio is based on JetBrains's IntelliJ. IntelliJ is a commercial Java IDE that also has a community (non-paid) version. This version serves as the base for Android Studio.

Setting Up Android Studio

Android Studio's version at the time of writing is 3.2.1; hopefully the version won't be very different by the time you read this book. You can download it from `https://developer.android.com/studio`. It's available for Windows (both 32- and 64-bit), macOS, and Linux. I ran the installation instructions on macOS (Mojave), Windows 10 64-bit, and Ubuntu 18. I work primarily in a macOS environment, which explains why most of the screen grabs for this book looks like macOS. Android Studio looks, runs, and feels (mostly) the same on all three platforms, with very minor differences like key bindings and the main menu bar in macOS.

Before we go further, let's look at the system requirements for Android Studio. At a minimum, you'll need the following:

- Microsoft Windows 7/8/10 (32 or 64-bit) or
- macOS 10.10 (Yosemite or higher) or
- Linux (Gnome or KDE Desktop), Ubuntu 14.04 or higher (64-bit but capable of running 32-bit applications)
- GNU C Library (glibc 2.19 or later) if you're on Linux

For hardware, your workstation needs to have at least

- 3GB RAM (8GB or more recommended)
- 2GB of available HDD space
- 1280 x 800 minimum screen resolution

This list came from the official Android website (`developer.android.com/studio`) and, of course, more is better. If you can snag 16GB RAM, 512GB SSD (or bigger), and a full HD (or UHD) monitor, that wouldn't be bad—not at all.

And now we get to the JDK (Java Development Kit) requirement. Starting with Android Studio 2.2, the installer comes with OpenJDK embedded. This way, a beginner programmer won't have to bother with the installation of a separate JDK, but you can still install a separate JDK if that's your preference. In this book, I'll assume that you will use the embedded OpenJDK that comes with Android Studio.

Download the installer from `https://developer.android.com/studio/` and get the proper binary file for your platform.

If you're in macOS, do the following:

1. Unpack the installer zipped file.
2. Drag the application file into the `Applications` folder.
3. Launch Android Studio.
4. Android Studio will prompt you to import some settings if you have a previous installation. You can import that—it's the default option.

> **Note** If you have an existing installation of Android Studio, you can keep using that version and still install Android Studio 3. It can coexist with your existing version of Android Studio because its settings will be kept in a different directory.

If you're in Windows, do the following:

1. Unzip the installer file.
2. Move the unzipped directory to a location of your choice, such as `C:\Users\myname\AndroidStudio`.
3. Drill down to the `AndroidStudio` folder. Inside it is the `studio64.exe` file. This is the file you need to launch. It's a good idea to create a shortcut for this file. If you right-click `studio64.exe` and choose Pin to Start Menu, you can make Android Studio available from the Windows Start menu. Alternatively, you can also pin it to the taskbar.

The Linux installation requires a bit more work than simply double-clicking and following the installer prompts. In future releases of Ubuntu (and its derivatives), this might change and become as simple and frictionless as its Windows and macOS counterparts, but for now, you need to do some tweaking. The extra activities on Linux are mostly because AS needs some 32-bit libraries and hardware acceleration.

> **Note** The installation instructions in this section are meant for Ubuntu 64-bit and other Ubuntu derivatives (e.g. Linux Mint, Lubuntu, Xubuntu, Ubuntu MATE, etc.). I chose this distribution because it is a very common Linux flavor, so readers of this book will be using this distribution. If you are running a 64-bit version of Ubuntu, you will need to pull some 32-bit libraries in order for AS to function well.

To start pulling the 32-bit libraries for Linux, run the following commands in a terminal window:

```
sudo apt-get update && sudo apt-get upgrade -y
sudo dpkg --add-architecture i386
sudo apt-get install libncurses5:i386 libstdc++6:i386 zlib1g:i386
```

When all the prep work is done, you need to do the following:

1. Unpack the downloaded installer file. You can unpack the file using command line tools or the GUI tools. You can, for example, right-click the file and select the "Unpack here" option, if your file manager has it.

2. After unzipping the file, rename the folder to `AndroidStudio`.

3. Move the folder to a location where you have read, write, and execute privileges. Alternatively, you can also move it to `/usr/local/AndroidStudio`.

4. Open a terminal window, go to the `AndroidStudio/bin` folder, and run the `./studio.sh` command.

5. At first launch, Android Studio will ask if you want to import some settings. If you have installed a previous version of Android Studio, you may want to import those settings.

Configuring Android Studio

If this is the first time you've installed Android Studio, you may want to configure a couple of things before diving into coding work. In this section, I'll walk you through the following:

- Acquiring some software that you'll need in order to create programs that target specific versions of Android
- Making sure you have all the SDK tools you need
- And, optionally, changing the way you get updates

Launch the IDE if you haven't done so yet, and click the Configure option, as shown in Figure 1-1. Choose Preferences from the drop-down list.

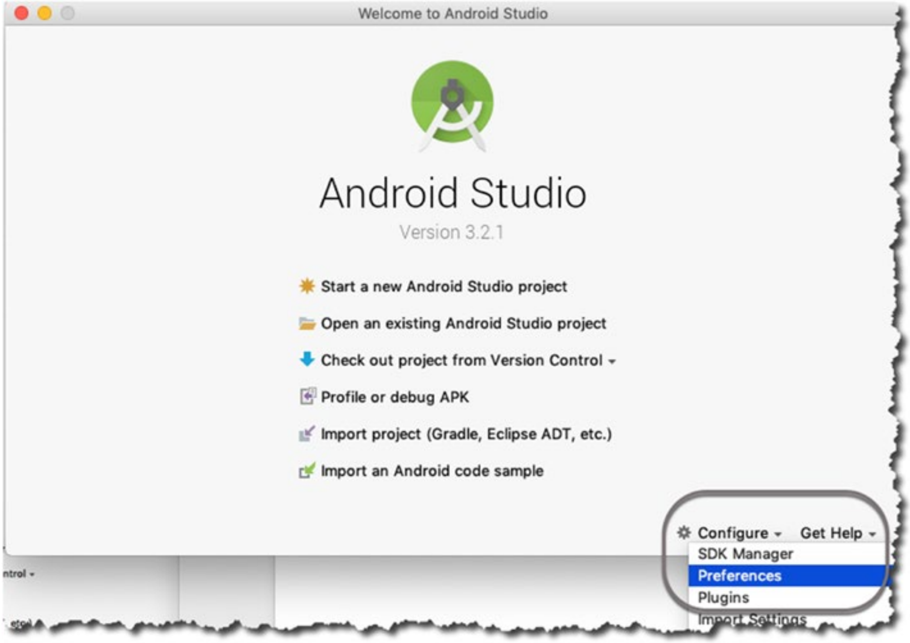
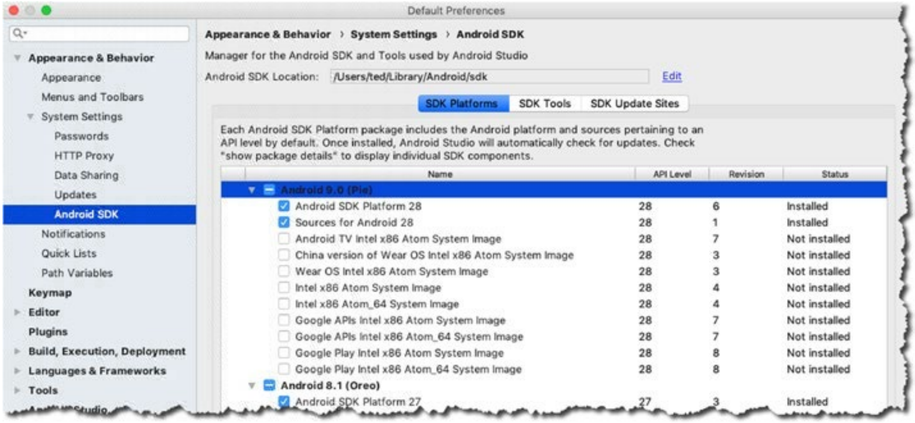

Welcome to Android Studio
Android Studio
Version 3.2.1
Start a new Android Studio project
Open an existing Android Studio project
Check out project from Version Control
Profile or debug APK
Import project (Gradle, Eclipse ADT, etc.)
Import an Android code sample
Configure
Get Help
SDK Manager
Preferences
Plugins
Import Settings

Figure 1-1. Go to Preferences from Android Studio's opening dialog

Clicking the Preferences option opens the Preferences dialog, as shown in Figure 1-2. On the left-hand side of the dialog, choose the Android SDK option.



Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: /Users/ted/Library/Android/sdk

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

Name	API Level	Revision	Status
▾ Android 9.0 (Pie)			
☑ Android SDK Platform 28	28	6	Installed
☑ Sources for Android 28	28	1	Installed
□ Android TV Intel x86 Atom System Image	28	7	Not installed
□ China version of Wear OS Intel x86 Atom System Image	28	3	Not installed
□ Wear OS Intel x86 Atom System Image	28	3	Not installed
□ Intel x86 Atom System Image	28	4	Not installed
□ Intel x86 Atom_64 System Image	28	4	Not installed
□ Google APIs Intel x86 Atom System Image	28	7	Not installed
□ Google APIs Intel x86 Atom_64 System Image	28	7	Not installed
□ Google Play Intel x86 Atom System Image	28	8	Not installed
□ Google Play Intel x86 Atom_64 System Image	28	8	Not installed
▾ Android 8.1 (Oreo)			
☑ Android SDK Platform 27	27	3	Installed

Figure 1-2. SDK platforms