

# Introduction

Create Your First Android App



Create your first  
Android app

This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](#)



# Contents

- Android Studio
- Creating "Hello World" app in Android Studio
- Basic app development workflow with Android Studio
- Running apps on virtual and physical devices



# Prerequisites

- Java Programming Language
- Object-oriented programming
- XML - properties / attributes
- Using an IDE for development and debugging

# Android Studio



Google Developers Training

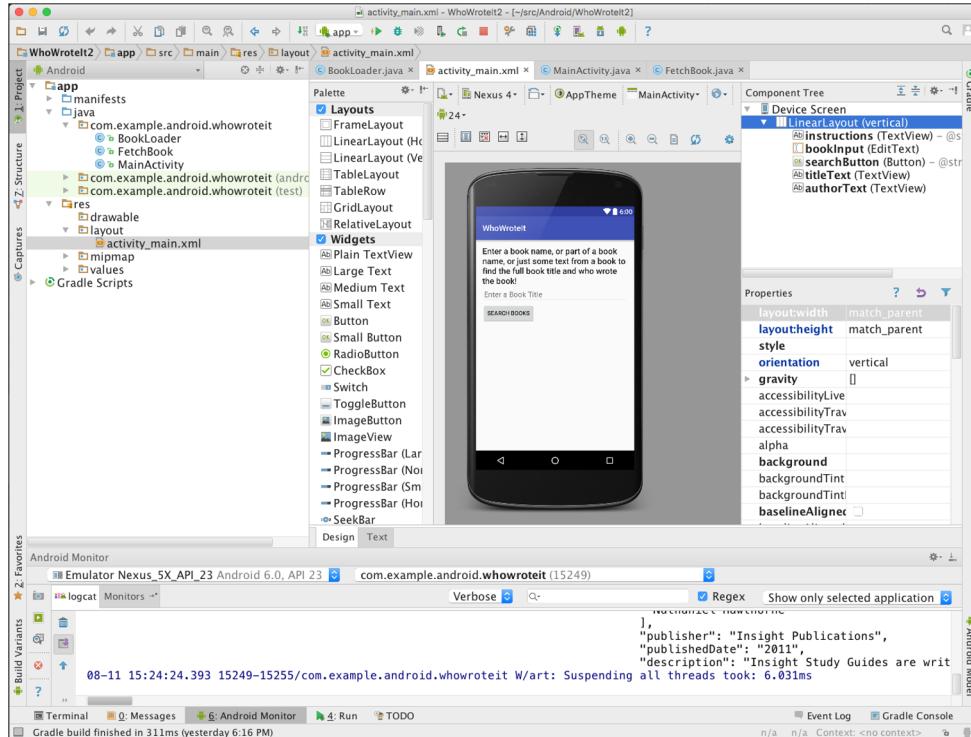
Android Developer Fundamentals

Create your first  
Android app

*This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](#)*



# What is Android Studio?



- Android IDE
- Project structure
- Templates
- Layout Editor
- Testing tools
- Gradle-based build
- Log Console
- Debugger
- Monitors
- Emulators

This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](#)



# Installation Overview

- Mac, Windows, or Linux
- Requires Java Development Kit (JDK) 1.7 or better from [Oracle Java SE downloads page](#)
- Set JAVA\_HOME to JDK installation location
- Download and install Android Studio from [http://developer.android.com/sdk/index.html](#)
- See [1.1 P Install Android Studio for details](#)

# Creating Your First Android App

# Start Android Studio



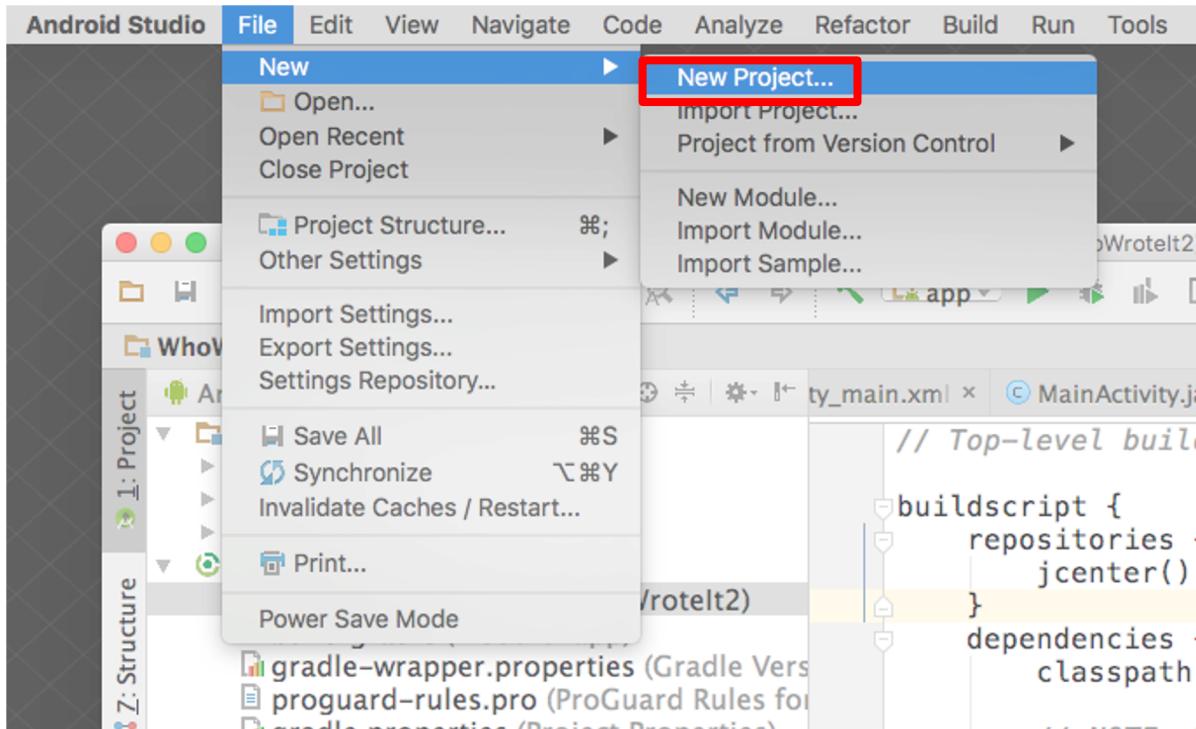
Welcome to Android Studio

Version 2.2 Beta 2 (AI-145.3200535)

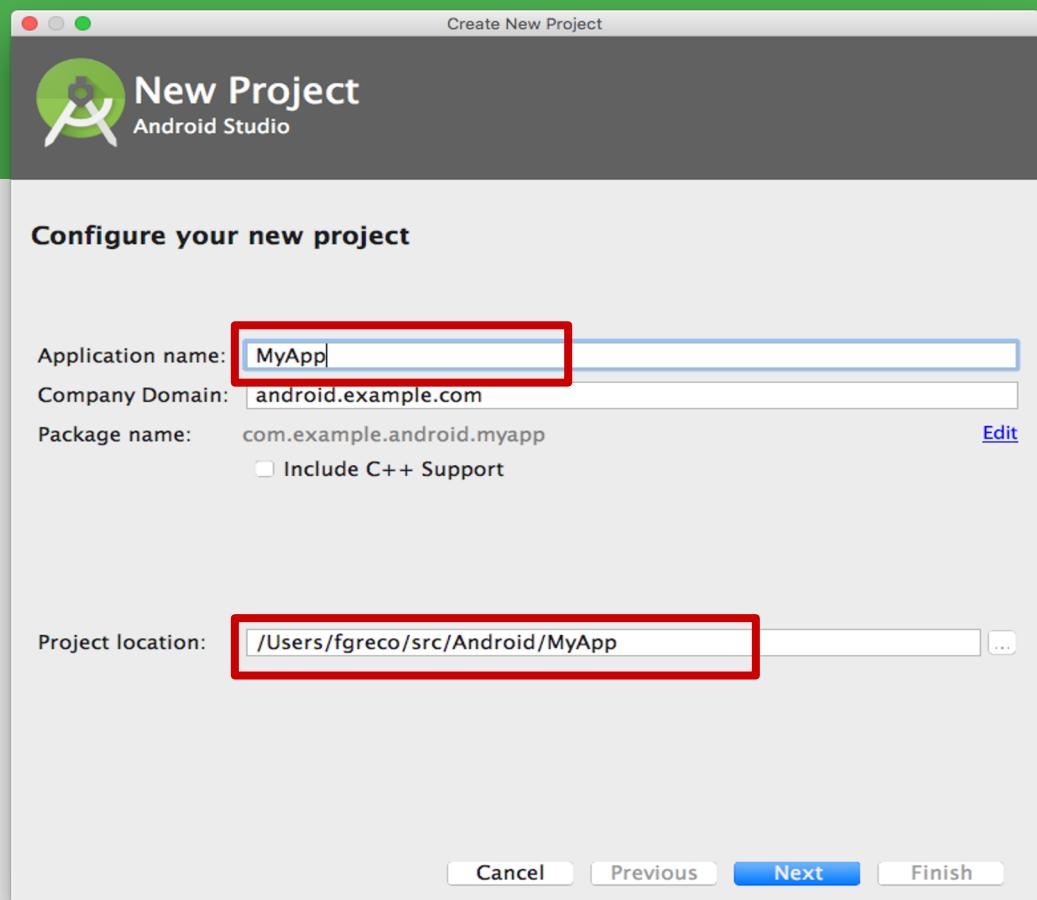
- Start a new Android Studio project**
- Open an existing Android Studio project
- Check out project from Version Control ▾
- Import project (Eclipse ADT, Gradle, etc.)
- Import an Android code sample

Configure ▾ Get Help ▾

# Create a project inside Android Studio



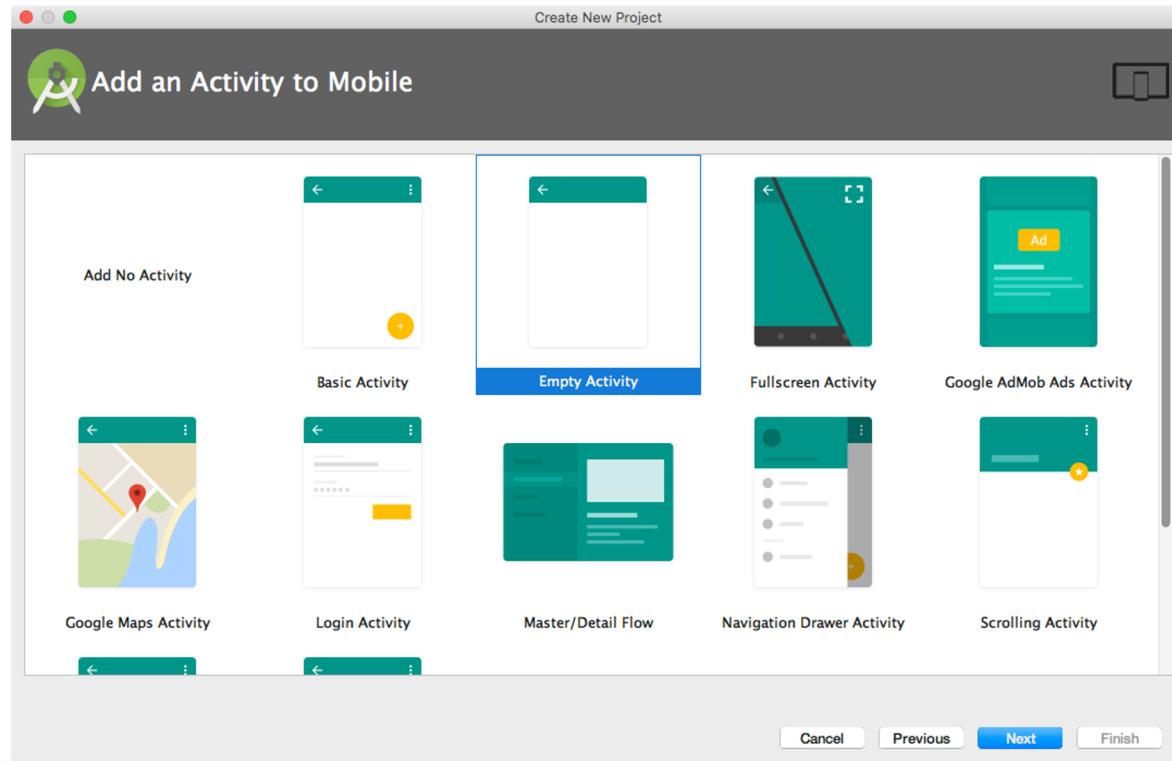
# Name your app



# Pick activity template

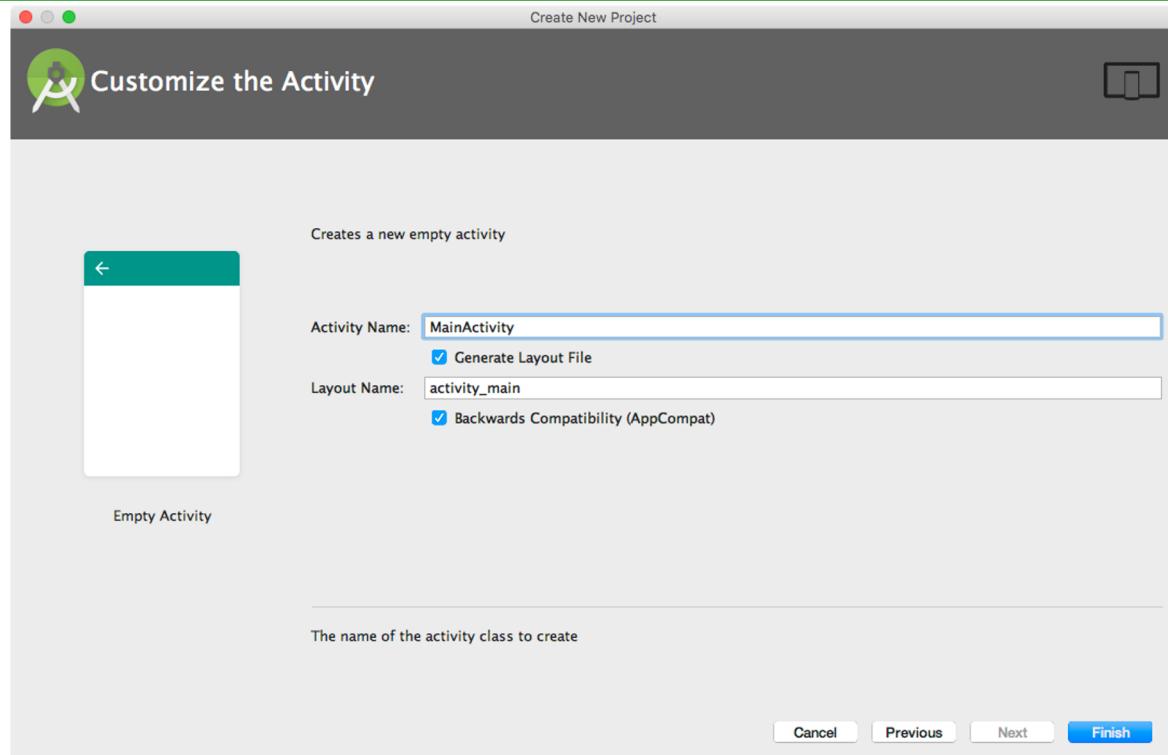
Choose templates for common activities, such as maps or navigation drawers.

Pick Empty Activity or Basic Activity for simple and custom activities.



# Name your activity

- Good practice to name main activity `MainActivity` and `activity_main` layout
- Use AppCompat
- Generating layout file is convenient



# Android Studio Panes

The screenshot shows the Android Studio interface with several open panes:

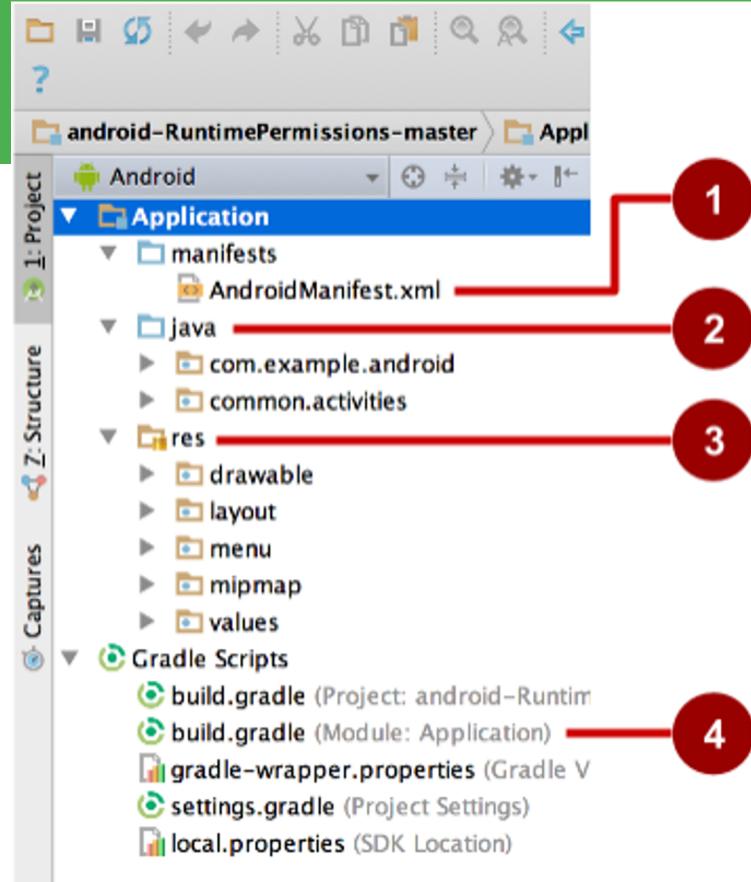
- Project Files**: A green box highlights the Project pane on the left, which displays the project structure for "HelloToast".
- Layout Editor**: A green box highlights the main workspace where the XML layout file `activity_main.xml` is being edited. The layout features a large yellow `TOAST` button with a blue number `0`, a `COUNT` button at the bottom, and a `show_count` TextView.
- Properties**: A green box highlights the Properties pane on the right, showing the current properties for the `show_count` TextView.
- Android Monitor**: A green box highlights the bottom pane, specifically the logcat tab, which displays log messages from the emulator.

**Android Monitors: logcat: log messages**

```
09-26 16:29:17.556 D/AndroidRuntime( 2724): Shutting down VM
09-26 16:29:17.620 D/dalvikvm( 2724): threadid=1: thread exiting with uncaught exception (group=0x416c5d00)
09-26 16:29:17.627 D/dalvikvm( 2724):   at com.example.android.hellotoast.MainActivity.onCreate(MainActivity.java:20)
09-26 16:29:17.642 D/dalvikvm( 2724):   at android.app.Activity.performCreate(Activity.java:5008)
09-26 16:29:17.642 D/dalvikvm( 2724):   at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1079)
09-26 16:29:17.642 D/dalvikvm( 2724):   at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2004)
09-26 16:29:17.642 D/dalvikvm( 2724):   at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2099)
09-26 16:29:17.642 D/dalvikvm( 2724):   at android.app.ActivityThread.access$700(ActivityThread.java:140)
09-26 16:29:17.642 D/dalvikvm( 2724):   at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1226)
09-26 16:29:17.642 D/dalvikvm( 2724):   at android.os.Handler.dispatchMessage(Handler.java:99)
09-26 16:29:17.642 D/dalvikvm( 2724):   at android.os.Looper.loop(Looper.java:137)
09-26 16:29:17.642 D/dalvikvm( 2724):   at android.app.ActivityThread.main(ActivityThread.java:4895)
09-26 16:29:17.642 D/dalvikvm( 2724):   at java.lang.reflect.Method.invokeNative(Native Method)
09-26 16:29:17.642 D/dalvikvm( 2724):   at java.lang.reflect.Method.invoke(Method.java:511)
09-26 16:29:17.642 D/dalvikvm( 2724):   at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:868)
09-26 16:29:17.642 D/dalvikvm( 2724):   at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:684)
09-26 16:29:17.642 D/dalvikvm( 2724):   at dalvik.system.NativeStart.main(Native Method)
09-26 16:29:17.645 I/dalvikvm( 2724): [D] connection established 0x7f06f9021c80, tid 1555
```

# Project folders

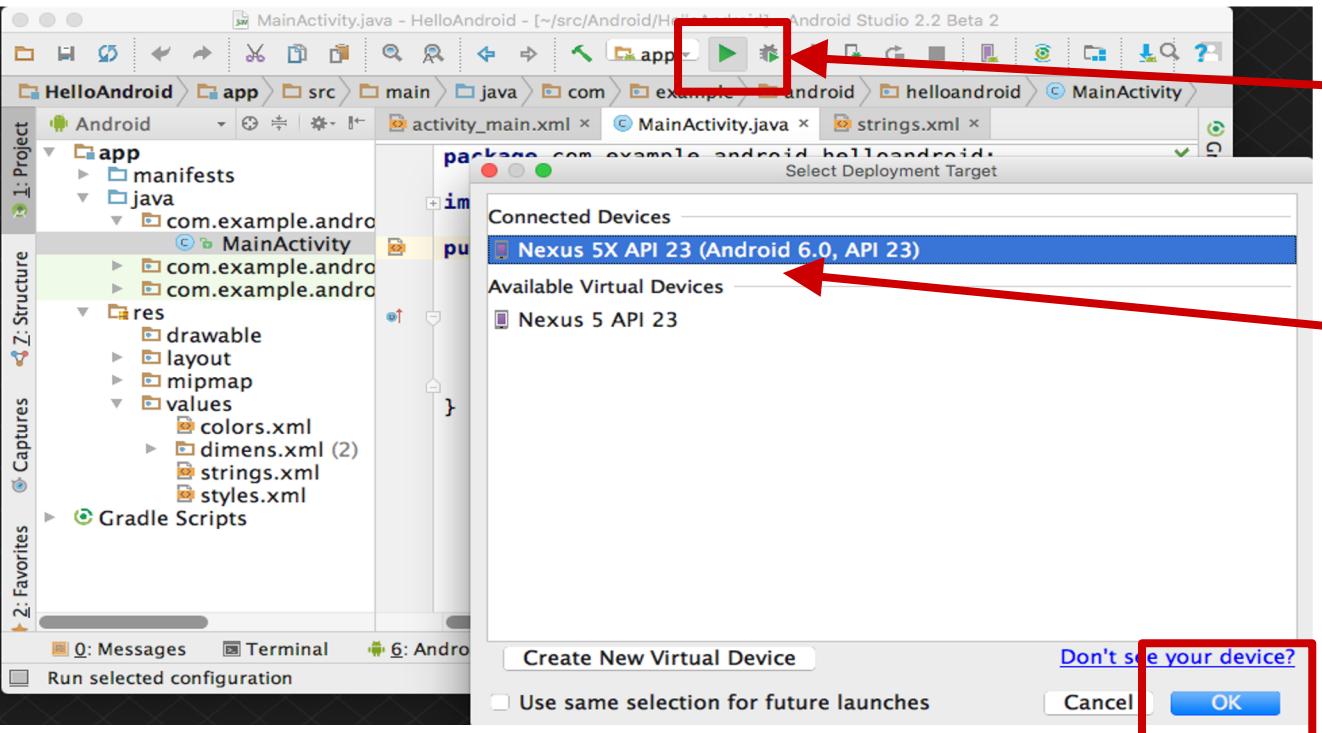
- 1. manifests**—Android Manifest file - description of app read by the Android runtime
- 2. java**—Java source code packages
- 3. res**—Resources (XML) - layout, strings, images, dimensions, colors...
- 4. build.gradle**—Gradle build files



# Gradle build system

- Modern build subsystem in Android Studio
- Three build.gradle:
  - project
  - module
  - settings
- Typically not necessary to know low-level Gradle details
- Learn more about gradle at <https://gradle.org/>

# Run your app



1. Run

2. Select virtual or physical device

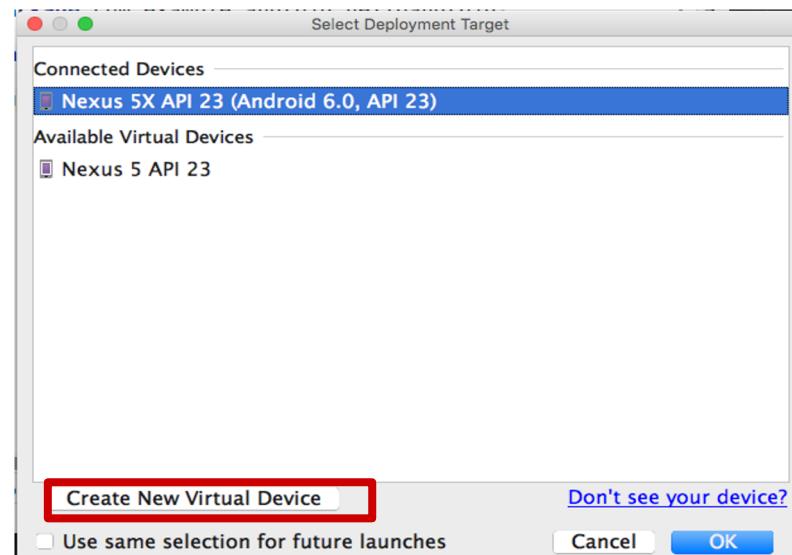
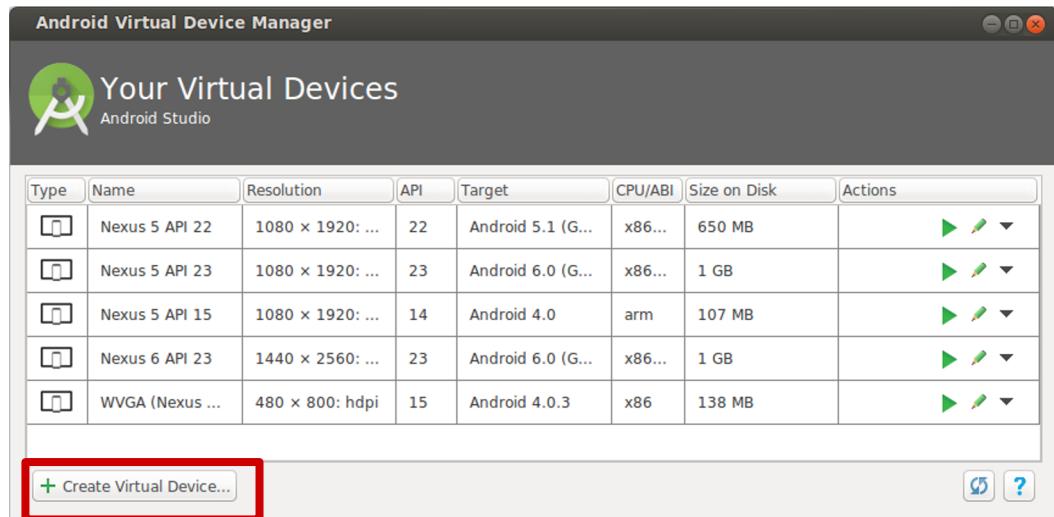
3. OK

# Create a virtual device

Use emulators to test app on different versions of Android and form factors.

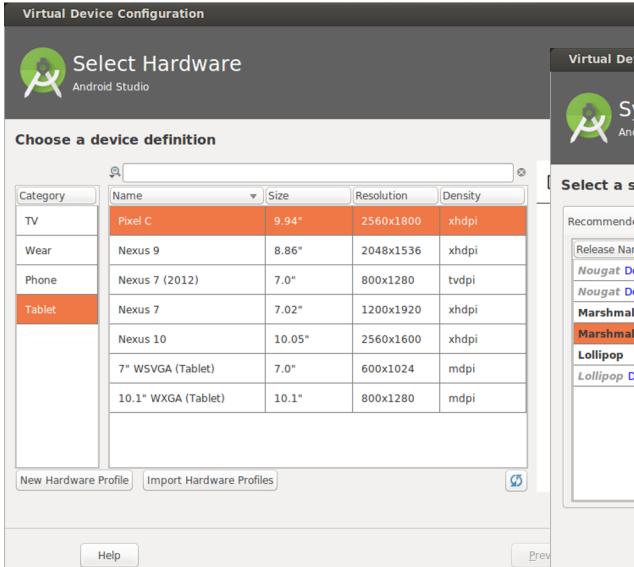
Tools > Android > AVD Manager

or:

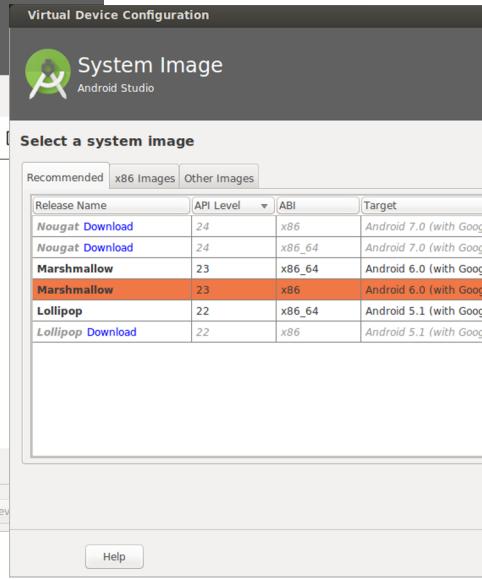


# Configure virtual device

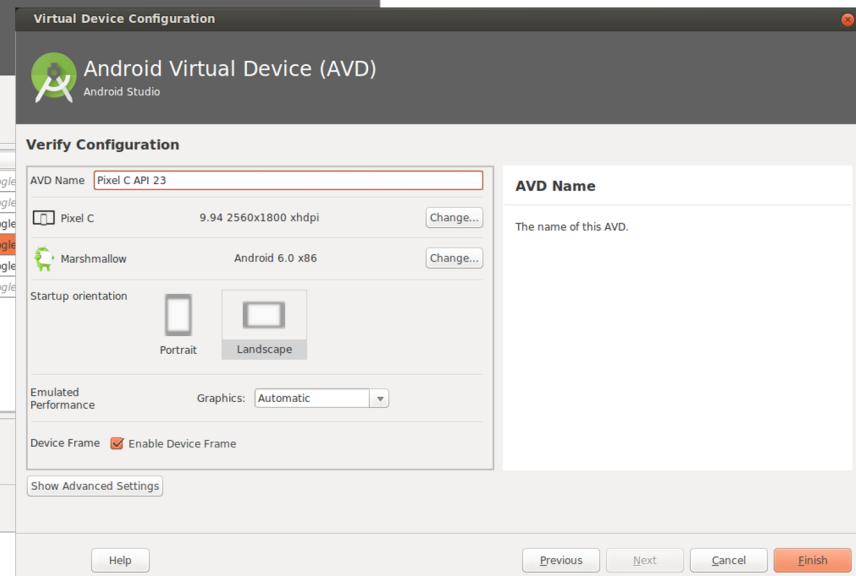
## 1. Choose hardware



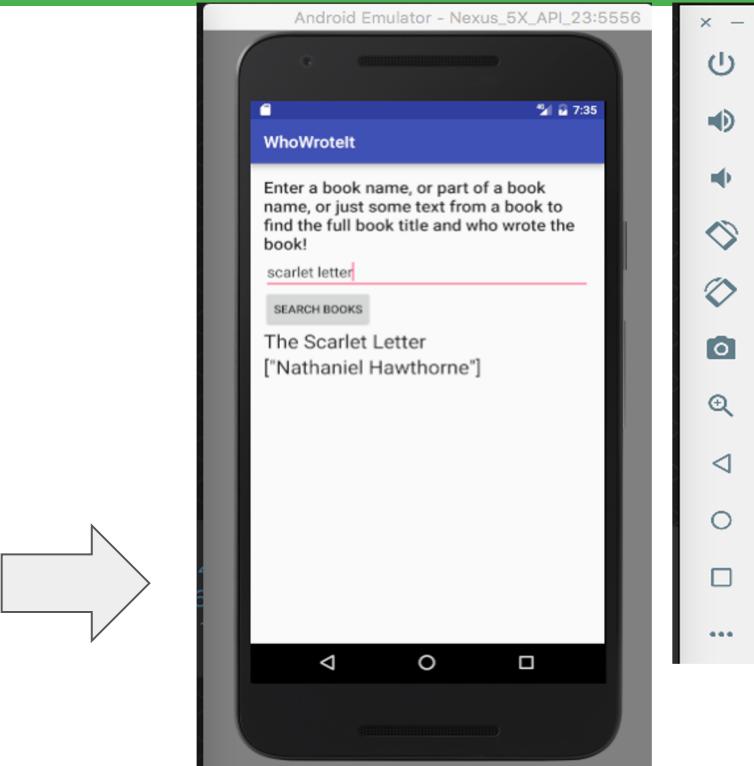
## 2. Select Android Version



## 3. Finalize



# Run on a virtual device



# Run on a physical device

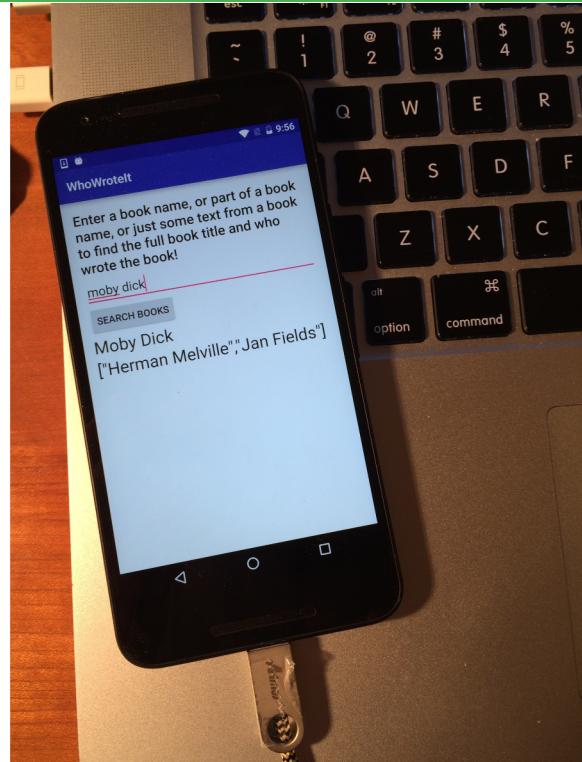
1. Turn on Developer Options:
  - a. **Settings > About phone**
  - b. Tap **Build number** seven times
2. Turn on USB Debugging
  - a. **Settings > Developer Options > USB Debugging**
3. Connect phone to computer with cable

Windows/Linux additional setup:

- [Using Hardware Devices](#)

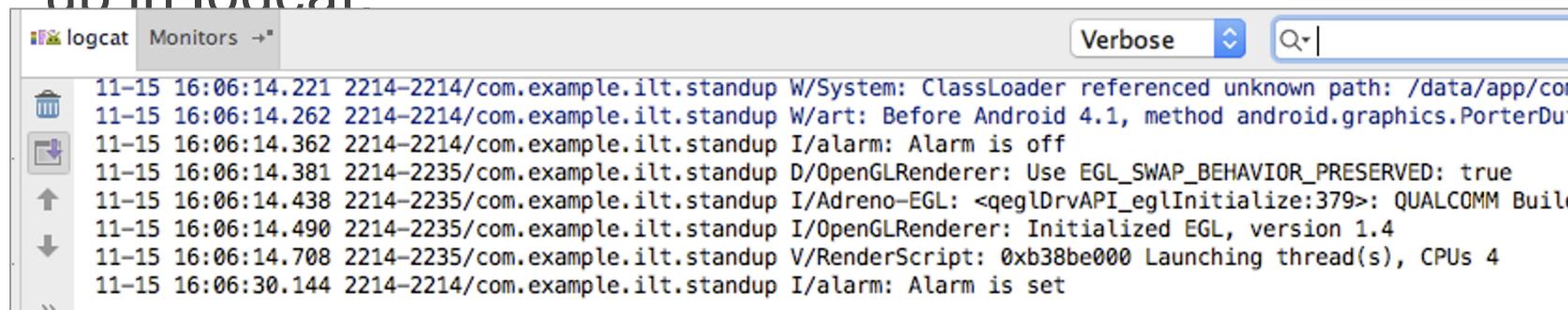
Windows drivers:

- [OEM USB Drivers](#)



# Get feedback as your app runs

- As the app runs, Android Monitor logcat shows information
- You can add logging statements to your app that will show up in logcat



The screenshot shows the Android Monitor's Logcat tab. The window has a toolbar at the top with 'logcat' selected, a dropdown for 'Monitors', and a search bar. On the left is a sidebar with icons for trash, download, upload, and refresh. The main area displays a list of log entries:

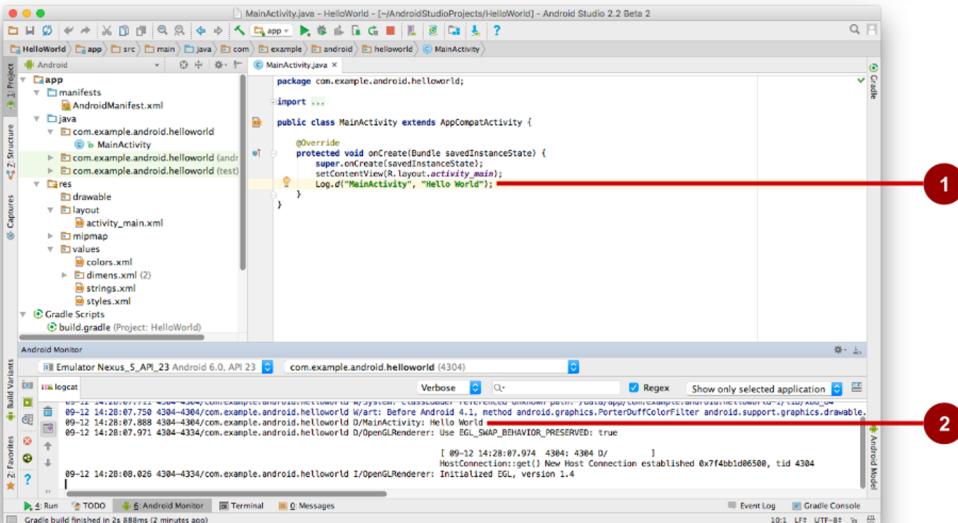
```
11-15 16:06:14.221 2214-2214/com.example.ilt.standup W/System: ClassLoader referenced unknown path: /data/app/com.example.ilt.standup-1/lib/arm64
11-15 16:06:14.262 2214-2214/com.example.ilt.standup W/art: Before Android 4.1, method android.graphics.PorterDuffColorFilter android.graphics.drawable.Drawable.setColorFilter(int,int) was defined as void in class android.graphics.drawable.Drawable. This violation has been allowed to keep APK size smaller but it's very likely that incompatibility might happen; fix it in code by either changing the type to Object or add an annotation (@TargetApi(16)) to the method declaration
11-15 16:06:14.362 2214-2214/com.example.ilt.standup I/Alarm: Alarm is off
11-15 16:06:14.381 2214-2235/com.example.ilt.standup D/OpenGLRenderer: Use EGL_SWAP_BEHAVIOR_PRESERVED: true
11-15 16:06:14.438 2214-2235/com.example.ilt.standup I/Adreno-EGL: <qeglDrvAPI_eglInitialize:379>: QUALCOMM Build ID: R665.1.0.0.0.0
11-15 16:06:14.490 2214-2235/com.example.ilt.standup I/OpenGLRenderer: Initialized EGL, version 1.4
11-15 16:06:14.708 2214-2235/com.example.ilt.standup V/RenderScript: 0xb38be000 Launching thread(s), CPUs 4
11-15 16:06:30.144 2214-2214/com.example.ilt.standup I/Alarm: Alarm is set
```

# Logging

```
import android.util.Log;  
  
// Use class name as tag  
private static final String TAG =  
    MainActivity.class.getSimpleName();  
  
// Show message in Android Monitor, logcat pane  
// Log.<log-level>(TAG, "Message");  
Log.d(TAG, "Creating the URI...");
```



# Android Monitor > logcat pane



1. Log statements in code.
2. logcat pane shows system and logging messages

- Set filters to see what's important to you
- Search using tags

# Learn more

- [Meet Android Studio](#)
- Official Android documentation at [developer.android.com](#)
- [Create and Manage Virtual Devices](#)
- [Supporting Different Platform Versions](#)
- [Supporting Multiple Screens](#)

# Learn even more

- [Gradle Wikipedia page](#)
- [Google Java Programming Language style guide](#)
- Find answers at [Stackoverflow.com](#)



# What's Next?

- Concept Chapter: [1.1 C Create Your First Android App](#)
- Practical: [1.1 P Install Android Studio and Run Hello World](#)

# END

