

Android Developer Fundamentals

Hello World

Lesson 1



1.2 Create Your First Android App

Contents

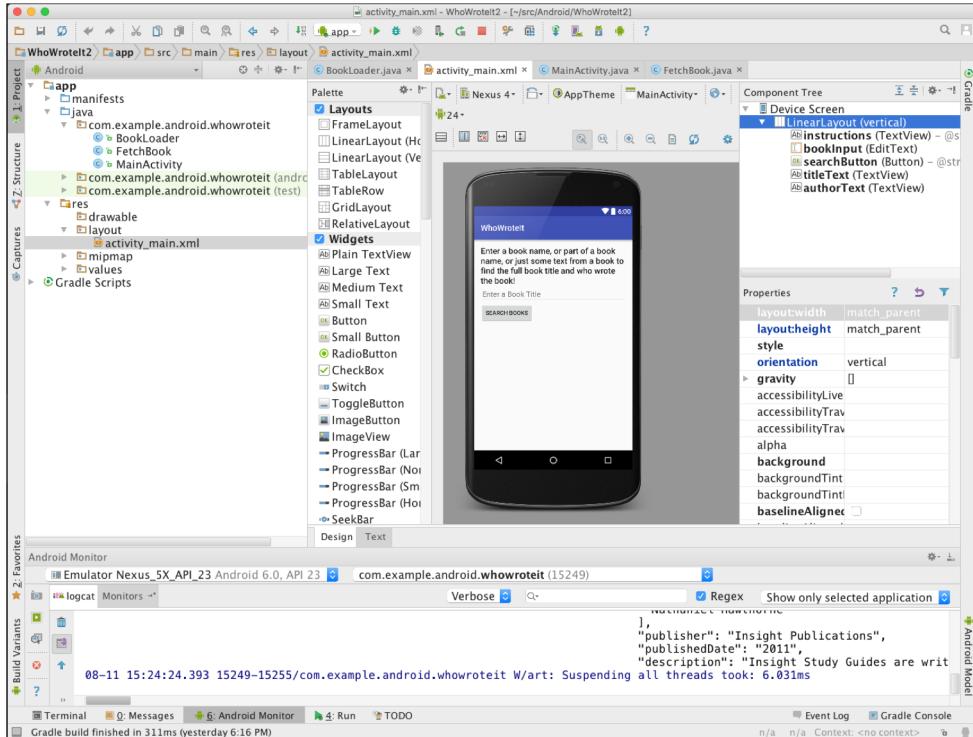
- Android Studio
- Creating "Hello World" app in Android Studio
- Basic app development workflow with Android Studio
- Running apps on virtual and physical devices

Prerequisites

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

Android Studio

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



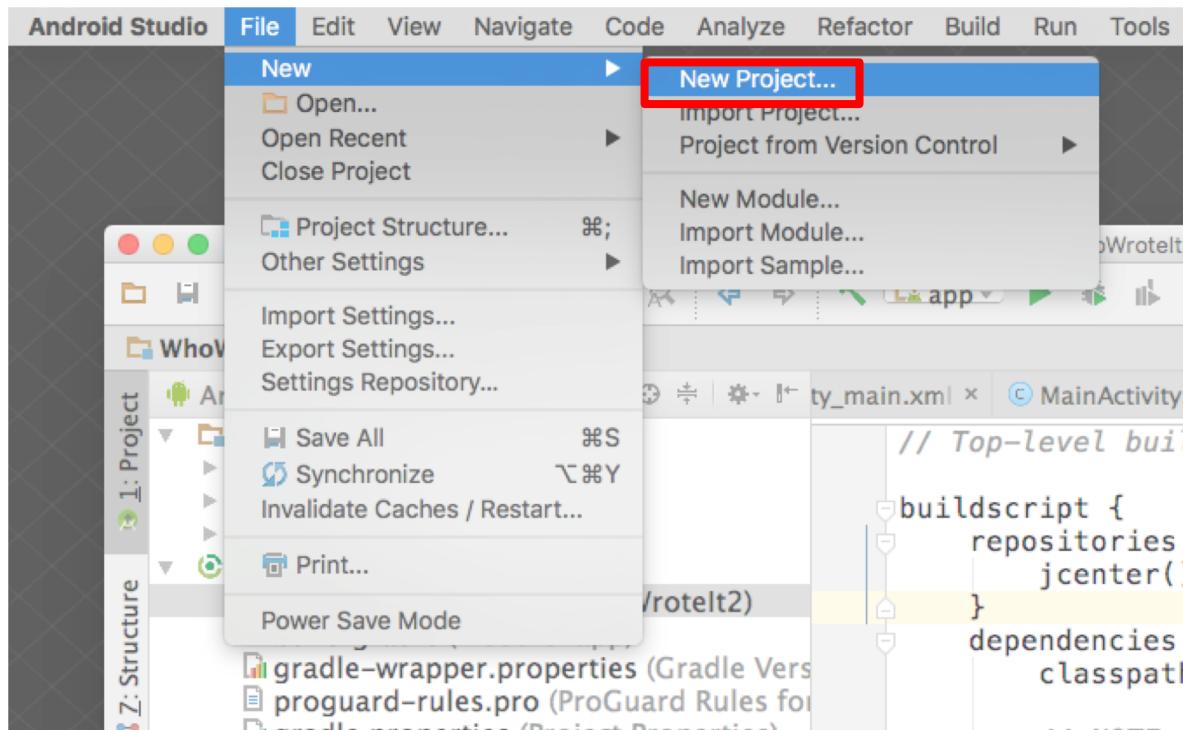
Welcome to Android Studio

The screenshot shows the "Welcome to Android Studio" screen. At the top center is the Android robot icon. Below it is the text "Android Studio" and "Version 2.2 Beta 2 (AI-145.3200535)". On the right side, there is a list of project creation options:

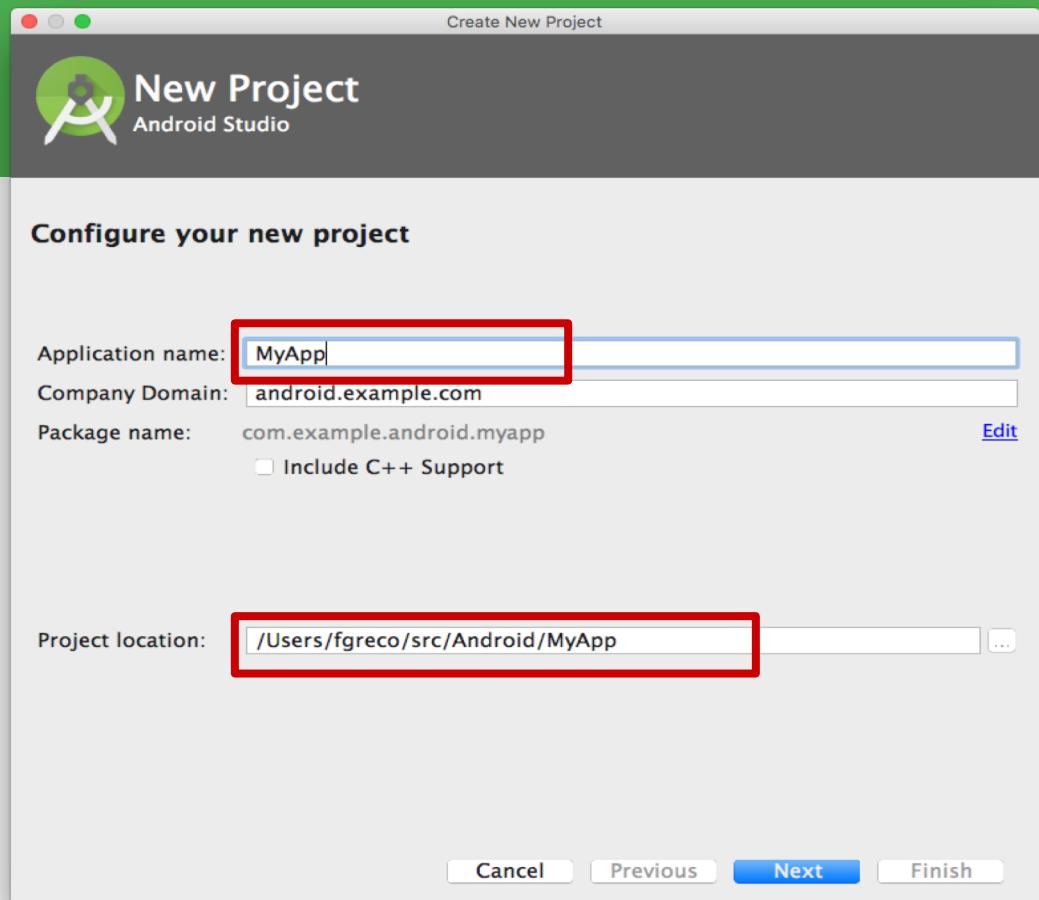
- Start a new Android Studio project** (highlighted with a red border)
- Open an existing Android Studio project
- Check out project from Version Control ▾
- Import project (Eclipse ADT, Gradle, etc.)
- Import an Android code sample

At the bottom right are "Configure ▾" and "Get Help ▾" buttons.

Create a project inside Android Studio



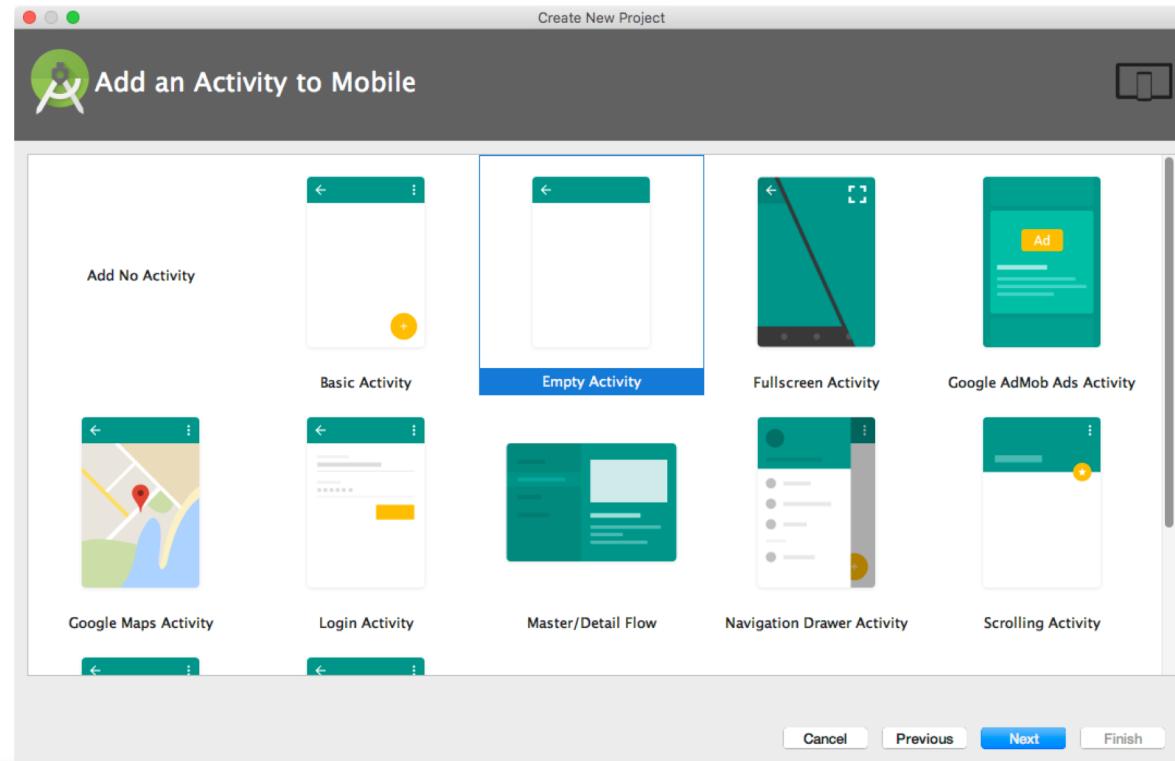
Name your app



Pick activity template

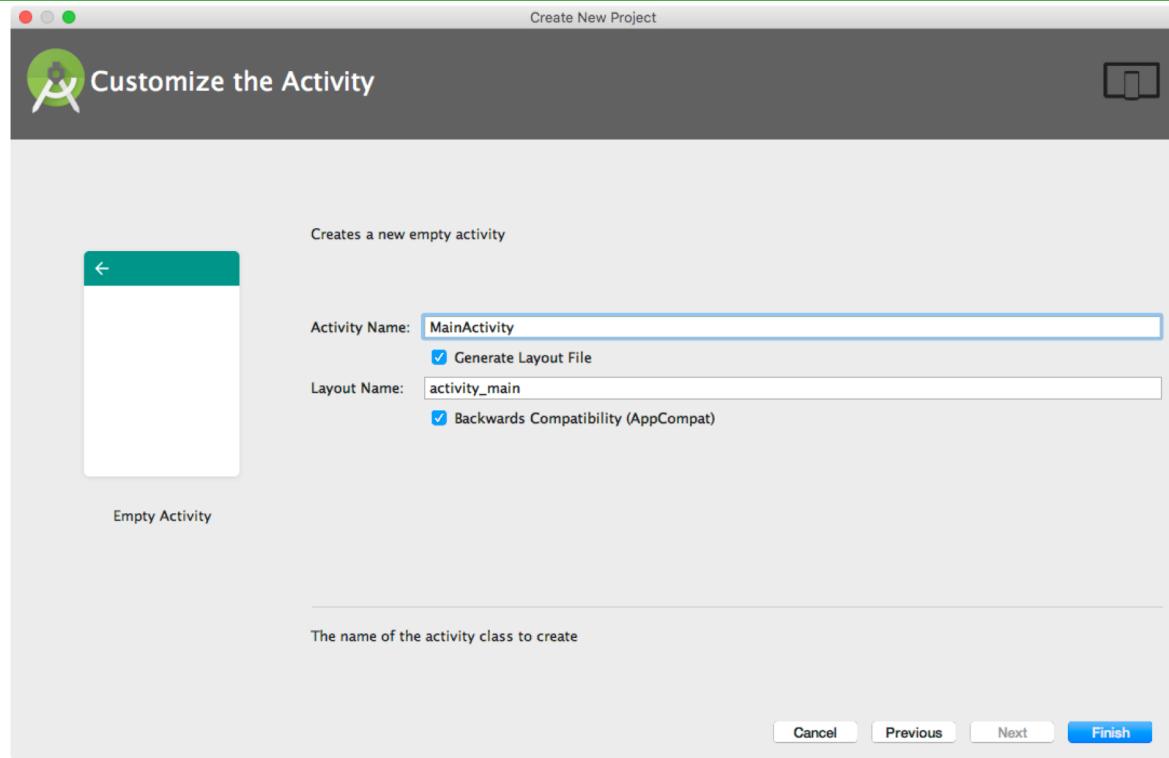
Choose templates for common activities, such as maps or navigation drawers.

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

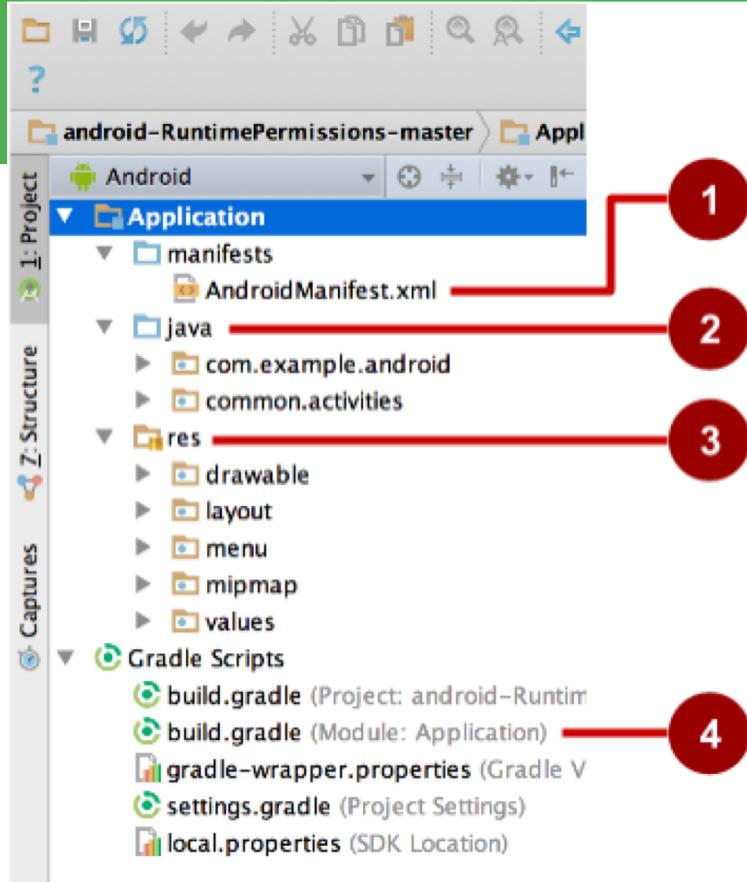
The screenshot illustrates the Android Studio interface with several open panes:

- Project Files**: Located on the left, showing the project structure under the `app` module.
- Layout Editor**: The central pane displays the `activity_main.xml` layout. It features a large yellow `LinearLayout` containing a `TextView` with the number "0". To its right is another `LinearLayout` with a `TextView` labeled "show_count" and a `Button` labeled "button_count". The `Component Tree` on the left lists the components: `LinearLayout (vertical)`, `button_toast`, `show_count`, and `button_count`.
- Android Monitors**: The bottom pane shows the `logcat` log for the emulator. The log output includes:

```
09-26 16:29:17.556 I/ActivityManager( 2724): Displayed com.example.android.hellotoast[0x10200]: +1ms
09-26 16:29:17.620 I/ActivityManager( 2724): Displayed com.example.android.hellotoast[0x10200]: +1ms
09-26 16:29:17.627 I/ActivityManager( 2724): Displayed com.example.android.hellotoast[0x10200]: +1ms
09-26 16:29:17.642 I/ActivityManager( 2724): Displayed com.example.android.hellotoast[0x10200]: +1ms
09-26 16:29:17.645 D/OpenGLRenderer( 2724): endAllActiveAnimators on 0x7f06f9021c80, tid 1555
    ] connection established 0x7f06f9021c80, tid 1555
```

Project folders

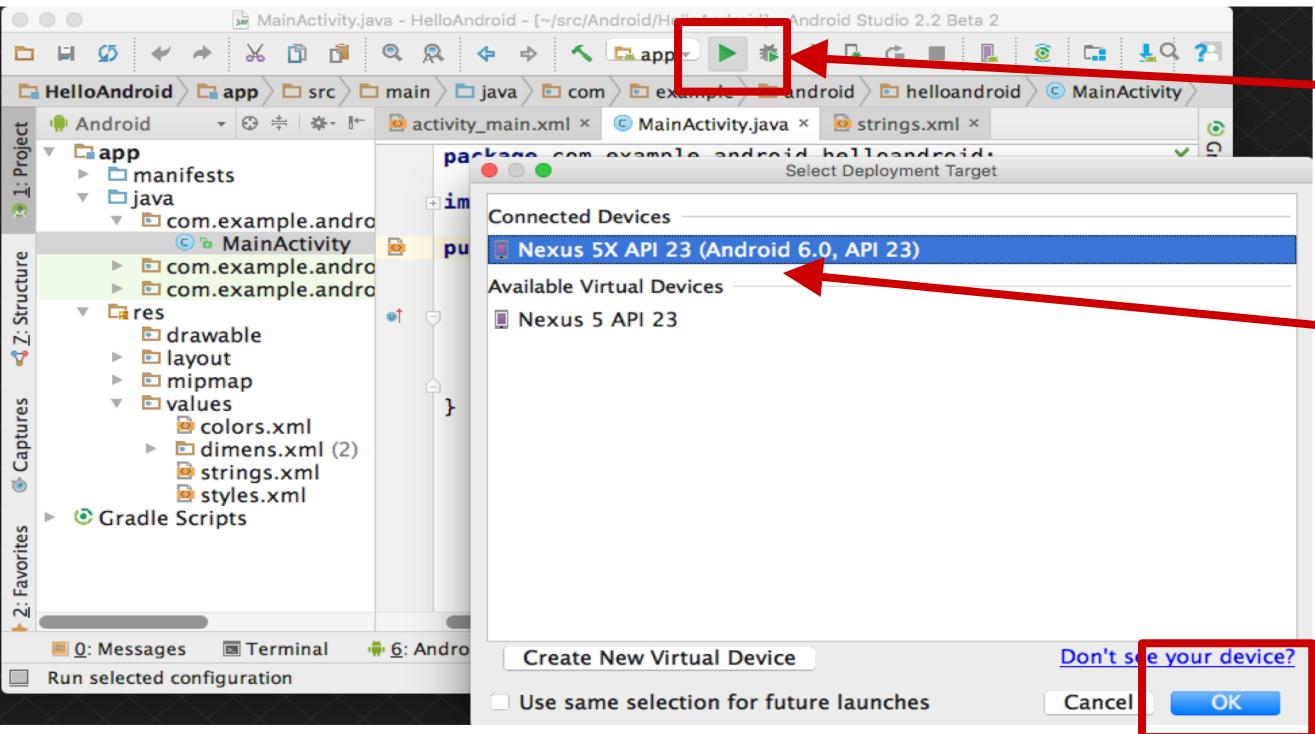
- 1. manifests**—Android Manifest file - description of app read by the Android runtime
- 2. java**—Java source code packages
- 3. res**—Resources (XML) - layout, strings, images, dimensions, colors...
- 4. build.gradle**—Gradle build files



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

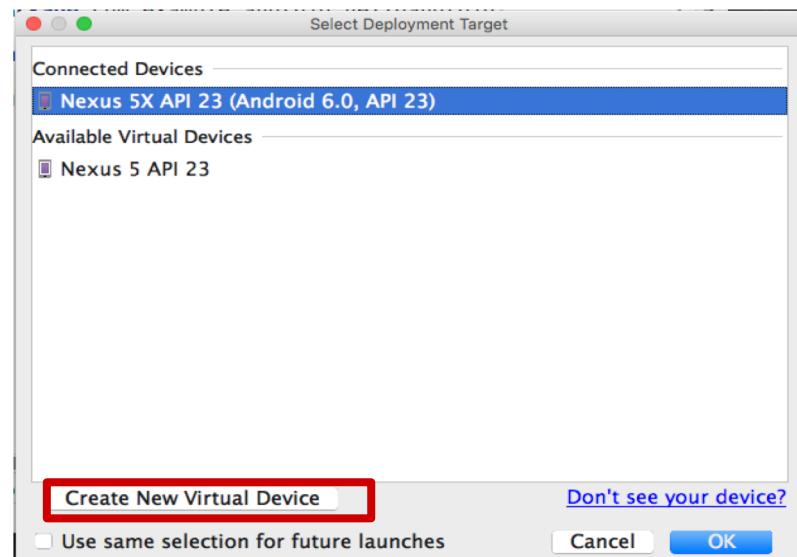
3. OK

Create a virtual device

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

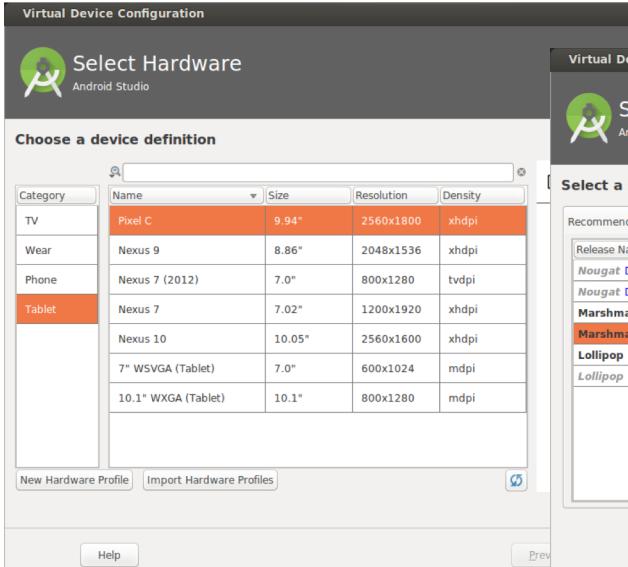
Tools > Android > AVD Manager

or:



Configure virtual device

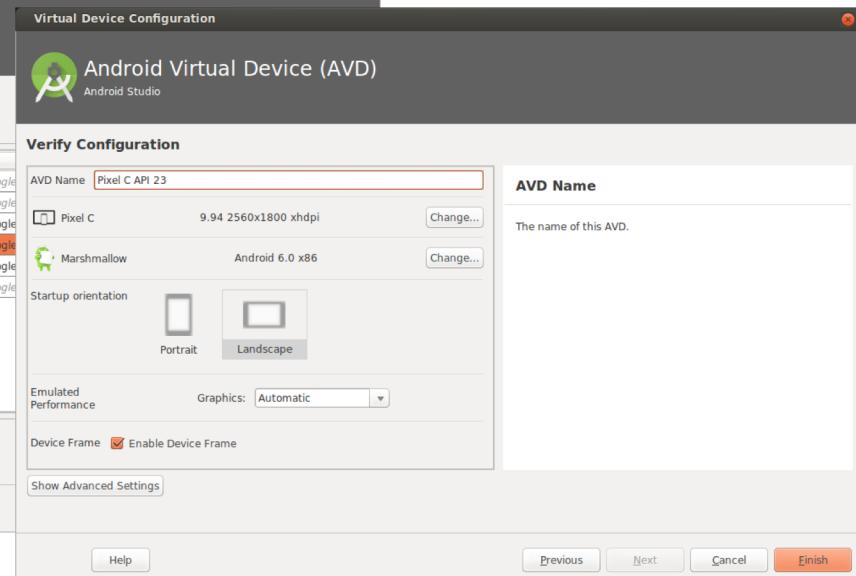
1. Choose hardware



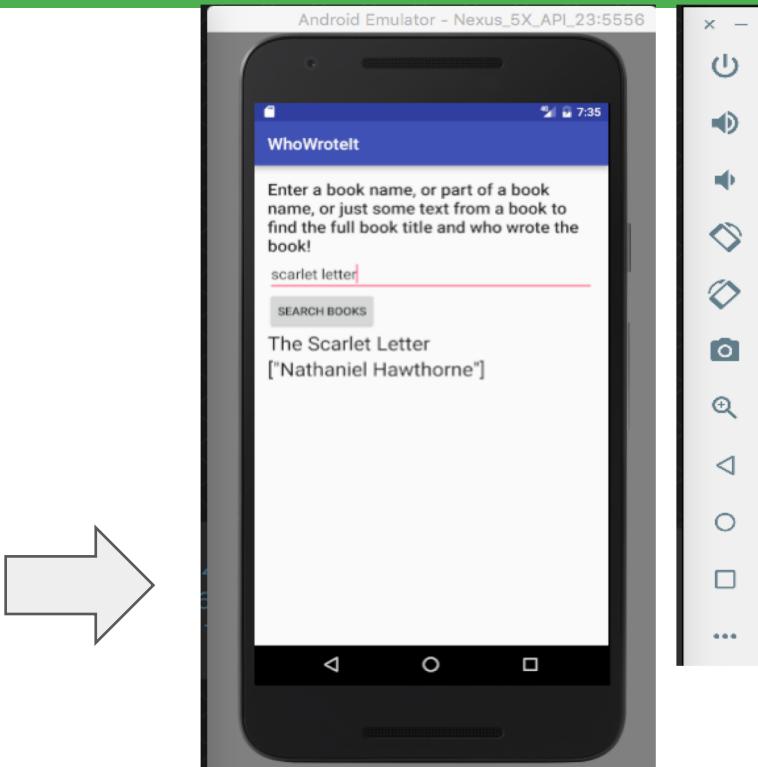
2. Select Android Version



3. Finalize



Run on a virtual device



Run on a physical device

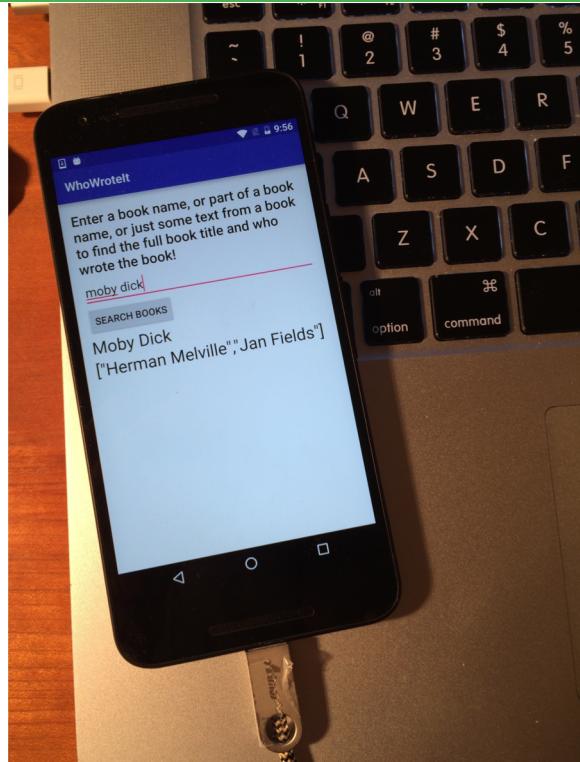
1. Turn on Developer Options:
 - a. **Settings > About phone**
 - b. Tap **Build number** seven times
2. Turn on USB Debugging
 - a. **Settings > Developer Options > USB Debugging**
3. Connect phone to computer with cable

Windows/Linux additional setup:

- [Using Hardware Devices](#)

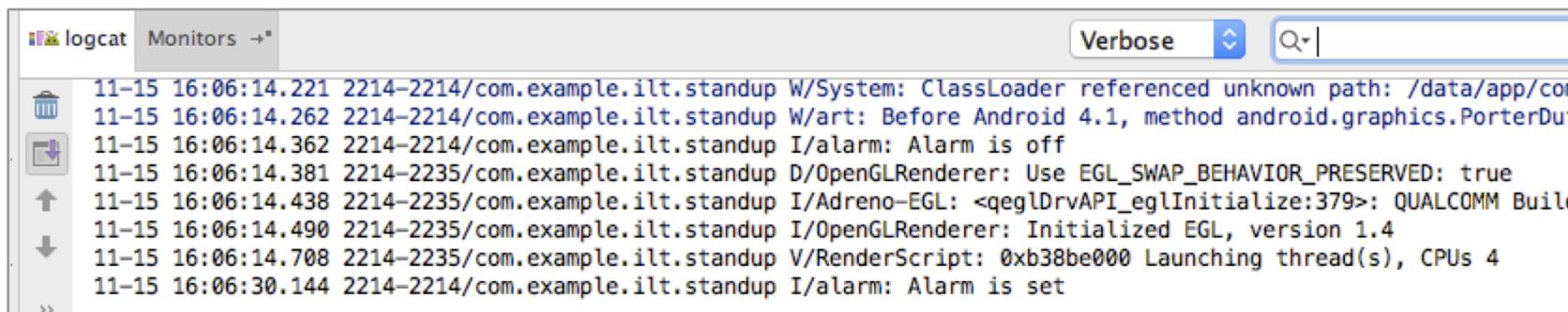
Windows drivers:

- [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.



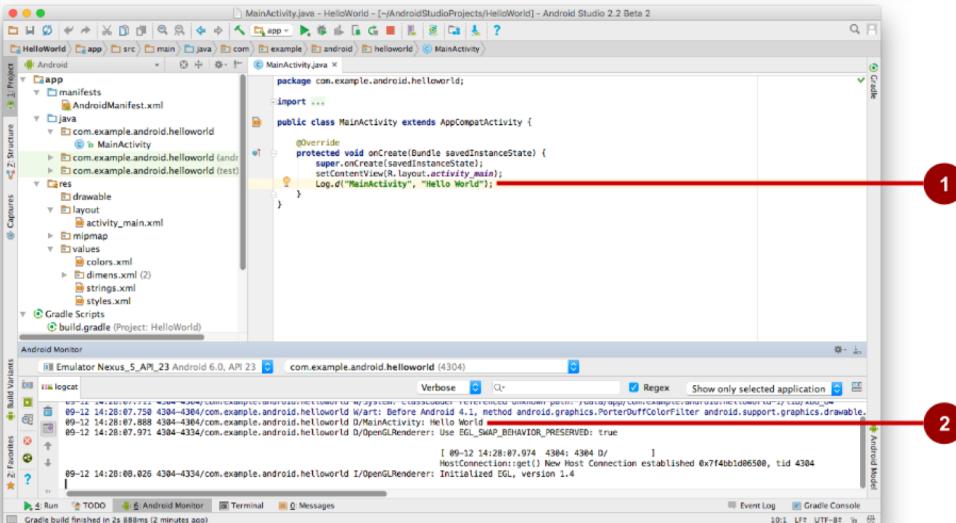
The screenshot shows the Android Monitor's Logcat tab. The window has a toolbar at the top with icons for logcat (highlighted), Monitors, a dropdown menu, and a search bar labeled 'Verbose'. The main area displays a list of log entries from an application named 'com.example.ilt.standup'. The log entries include various system and application logs, such as class loading, OpenGL renderer initialization, and alarm settings. The log entries are timestamped and show the package name, log level, timestamp, and the log message.

Time	Level	File	Message
11-15 16:06:14.221	W/System:	ClassLoader	referenced unknown path: /data/app/com.example.ilt.standup
11-15 16:06:14.262	W/art:	Before Android 4.1,	method android.graphics.PorterDuffXfermode
11-15 16:06:14.362	I/Alarm:	Alarm is off	
11-15 16:06:14.381	D/OpenGLRenderer:	Use EGL_SWAP_BEHAVIOR_PRESERVED:	true
11-15 16:06:14.438	I/Adreno-EGL:	<qeglDrvAPI_eglInitialize:379>: QUALCOMM Build	
11-15 16:06:14.490	I/OpenGLRenderer:	Initialized EGL, version 1.4	
11-15 16:06:14.708	V/RenderScript:	0xb38be000 Launching thread(s), CPUs 4	
11-15 16:06:30.144	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

