

Android Developer Fundamentals

Hello World

Lesson 1



1.1 Create Your First Android App



Contents

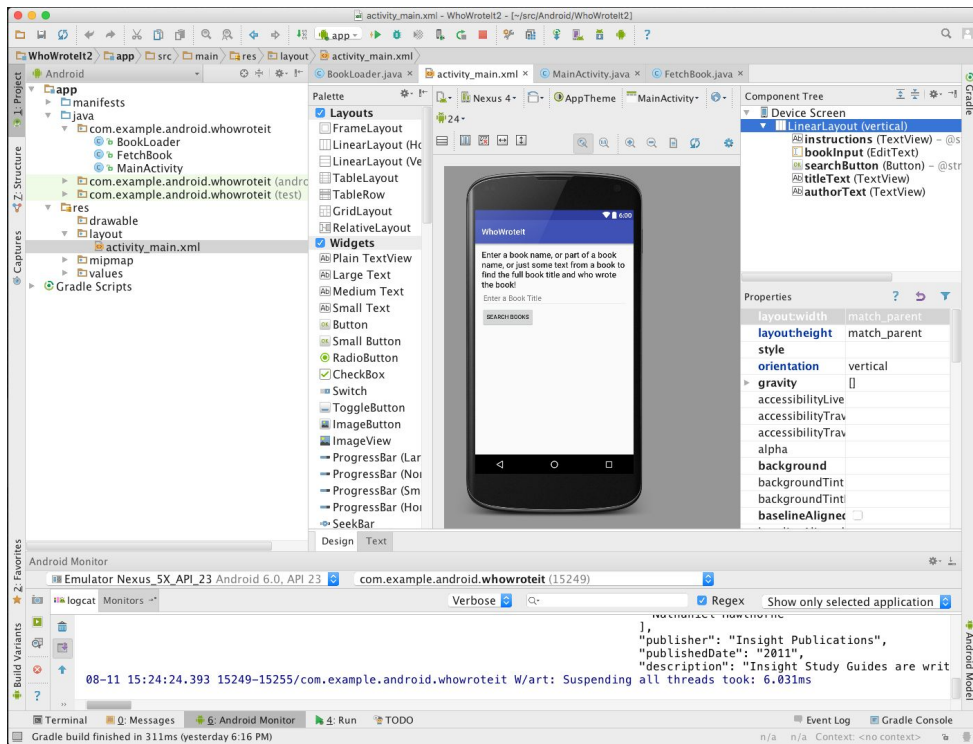
- Android Studio
- Creating "Hello World" app in Android Studio
- Basic app development workflow with Android Studio
- Running apps on virtual and physical devices

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

Installation Overview

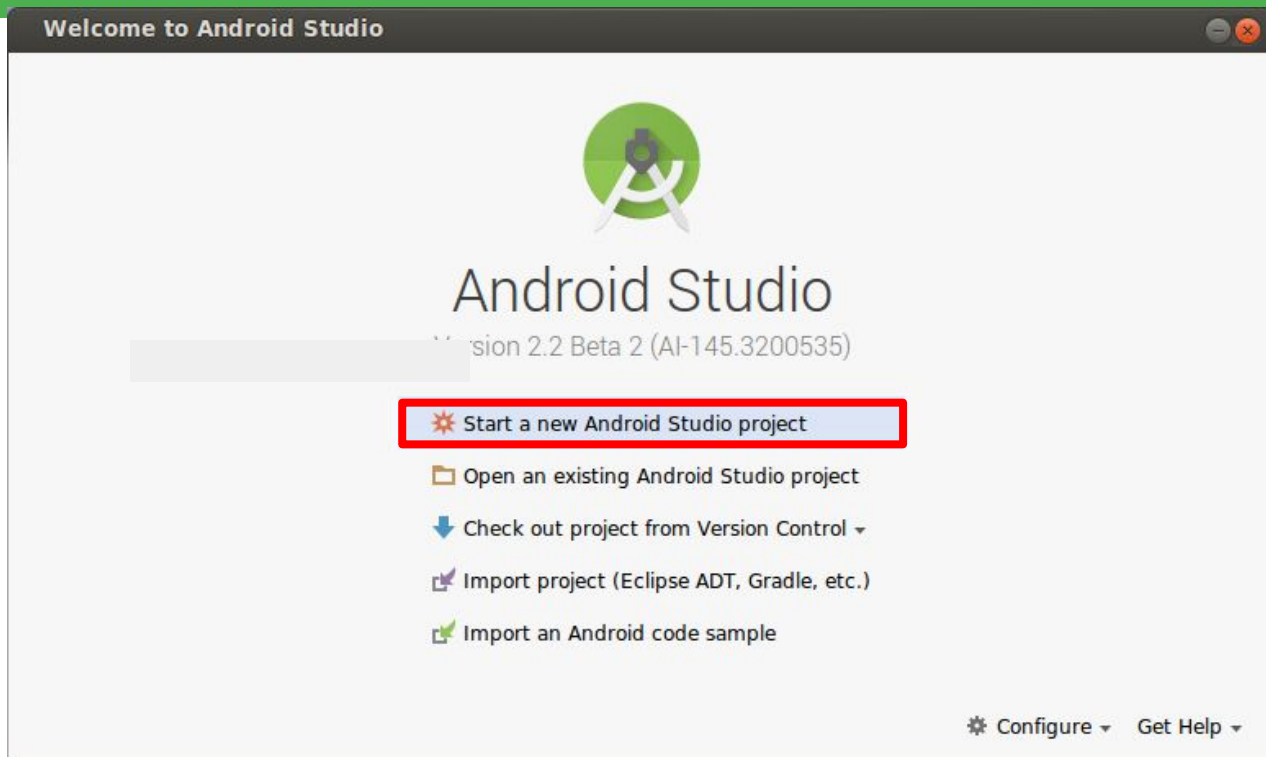
- Mac, Windows, or Linux
- Requires Java Development Kit (JDK) 1.7 or better from [Oracle Java SE downloads page](#)
- Set JAVA_HOME to JDK installation location
- Download and install Android Studio from <http://developer.android.com/sdk/index.html>
- See [1.1 P Install Android Studio for details](#)



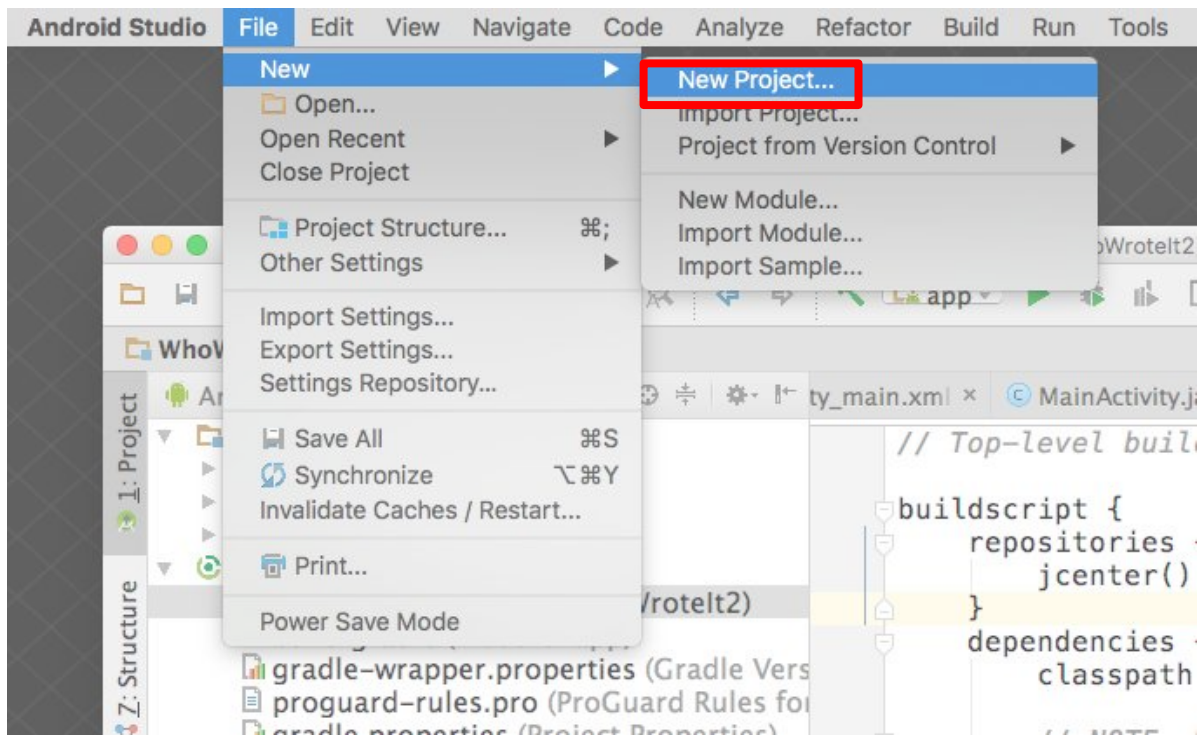
Creating Your First Android App



Start Android Studio




Create a project inside Android Studio



Name your app

Create New Project

 **New Project**
Android Studio

Configure your new project

Application name:

Company Domain:

Package name: [Edit](#)

☐ Include C++ Support

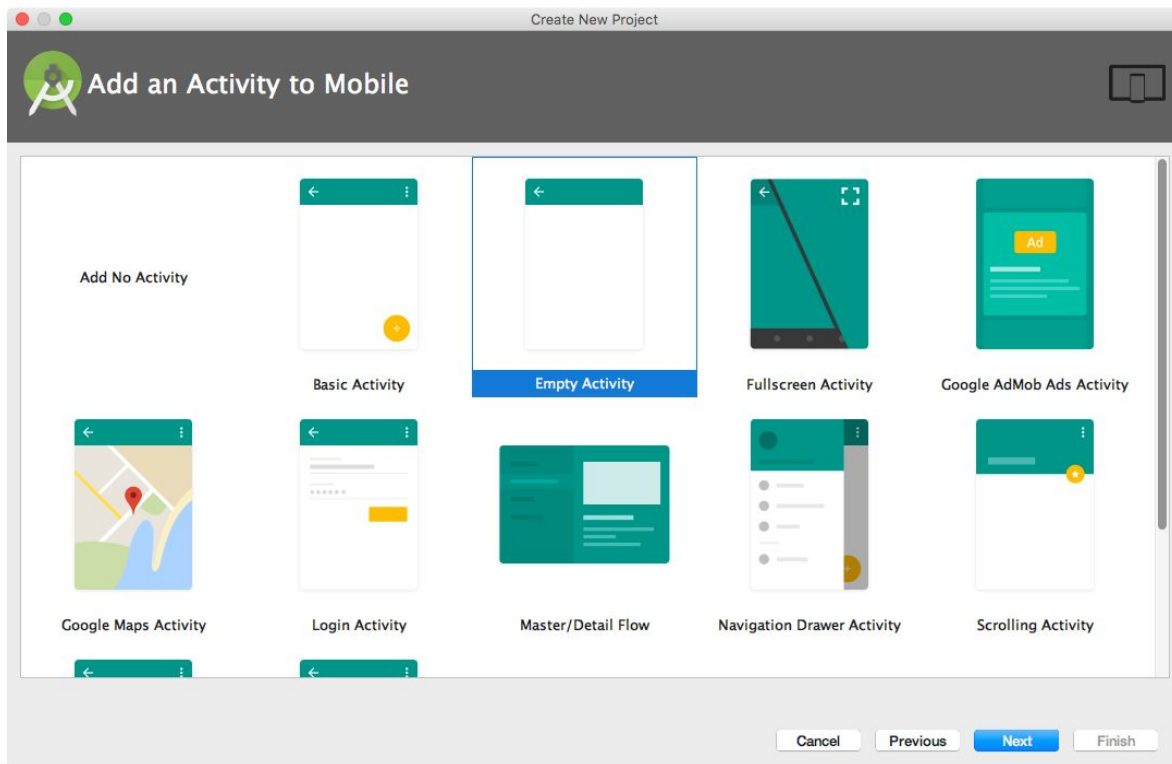
Project location:



Pick activity template

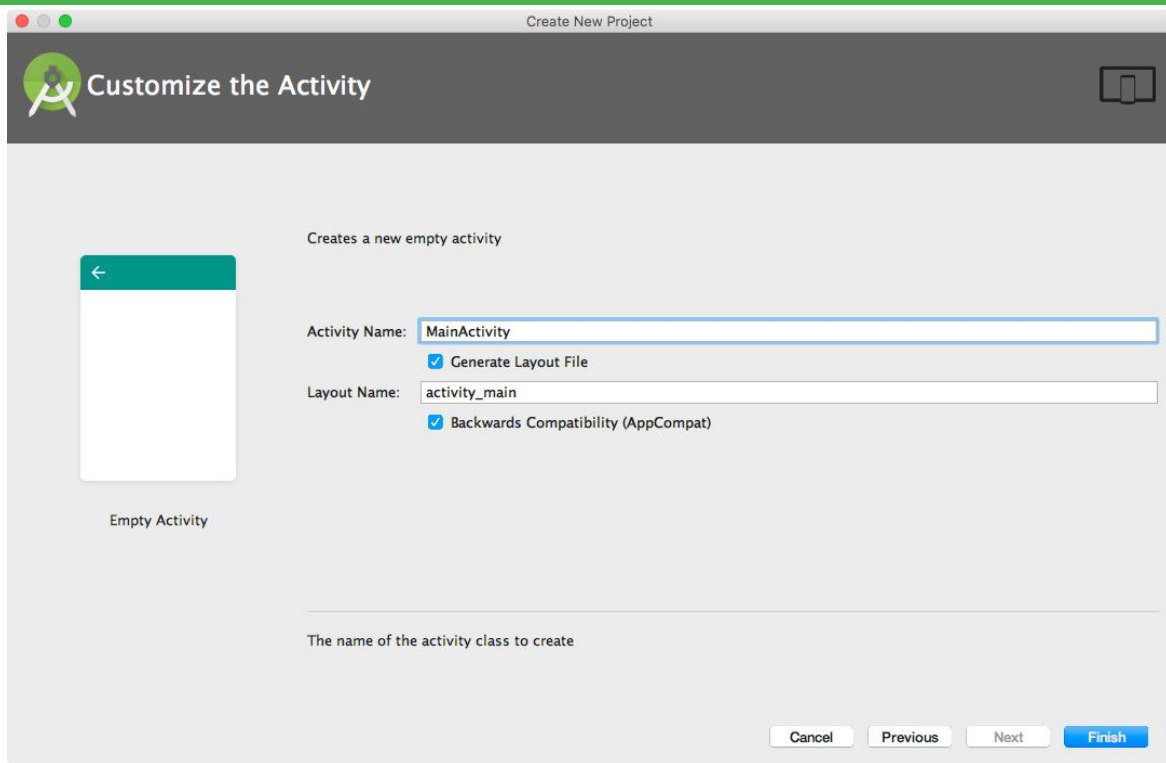
Choose templates for common activities, such as maps or navigation drawers.

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 AppCompatActivity
- Generating layout file is convenient



Android Studio Panes

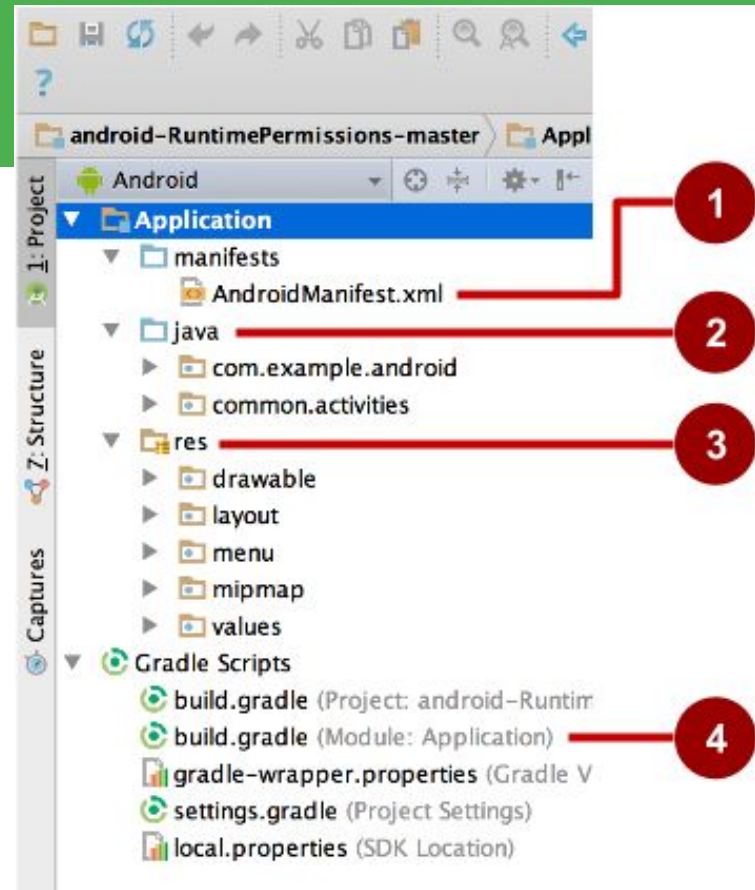
The screenshot displays the Android Studio interface with several key panes highlighted by green boxes and labels:

- Project Files:** Located on the left, it shows the project's file structure, including `manifests`, `java` (with `MainActivity`), `res` (with `layout` and `activity_main.xml`), `mipmap`, `values`, and `Gradle Scripts`.
- Layout Editor:** The central workspace for designing the UI. It features a visual editor with a yellow background and a large blue '0', and a corresponding XML view on the right showing the layout structure with components like `TextView` and `Button`. A **Component Tree** on the left lists the UI elements: `LinearLayout (vertical)`, `button_toast`, `show_count (TextView)`, and `button_count`.
- Android Monitors:** The bottom section, which includes the **logcat** window for viewing log messages (e.g., `09-26 16:29:17.556`), the **Terminal**, and the **Run** console.

Other visible elements include the top menu bar (File, Edit, View, etc.), the **Properties** panel on the right for the selected `show_count` TextView, and the **Android Monitor** header showing the selected application `com.example.android.hellotoast`.

Project folders

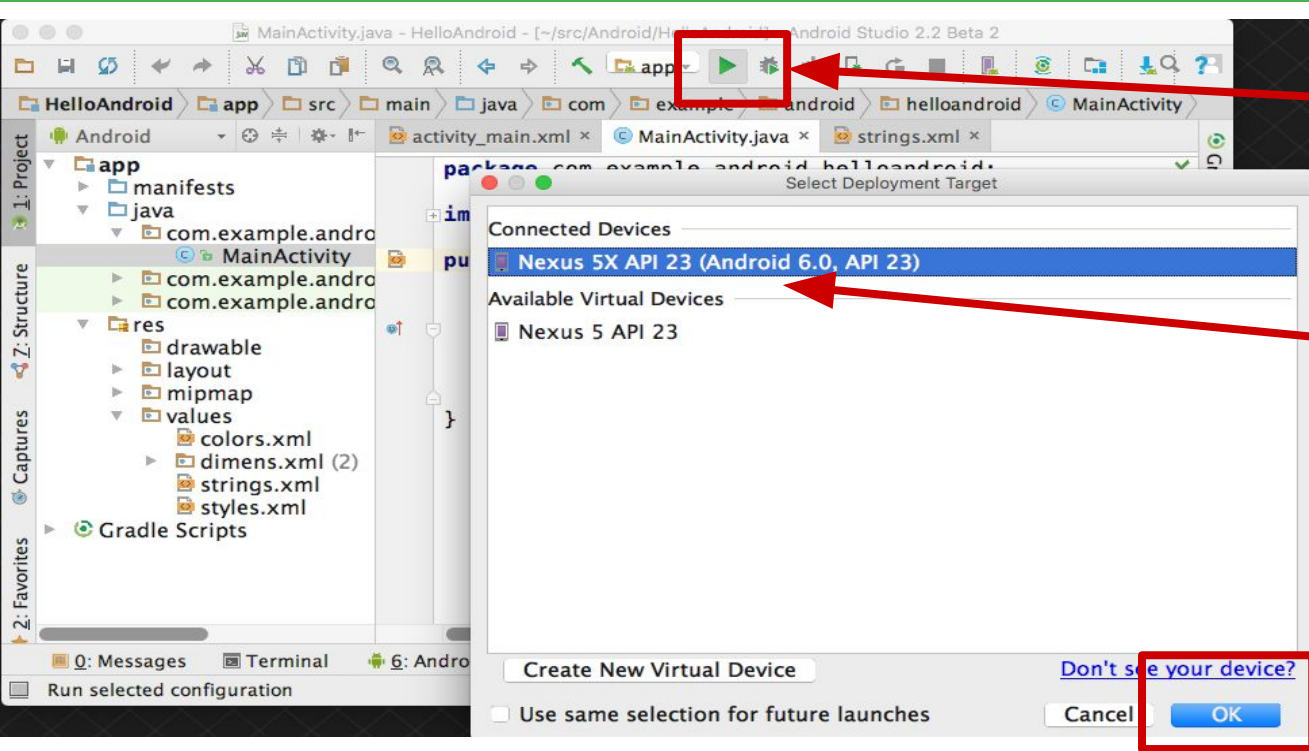
1. **manifests**—Android Manifest file - description of app read by the Android runtime
2. **java**—Java source code packages
3. **res**—Resources (XML) - layout, strings, images, dimensions, colors...
4. **build.gradle**—Gradle build files



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

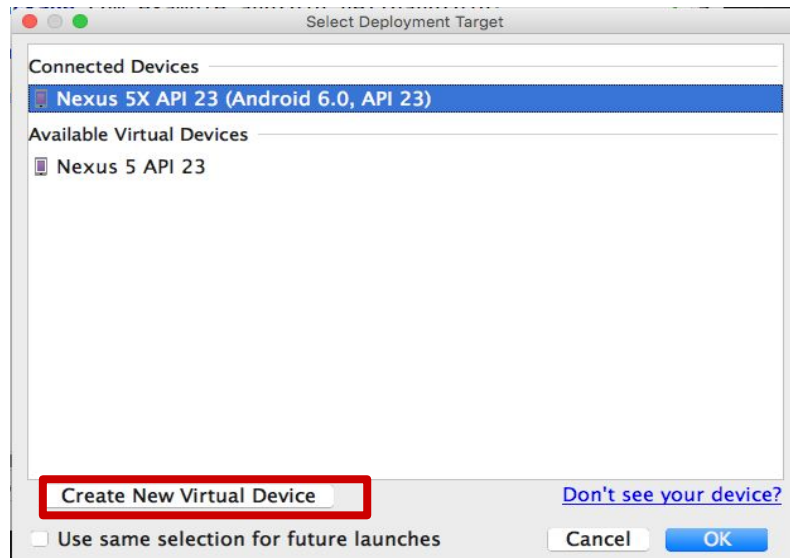
3. OK

Create a virtual device

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

Tools > Android > AVD Manager

or:

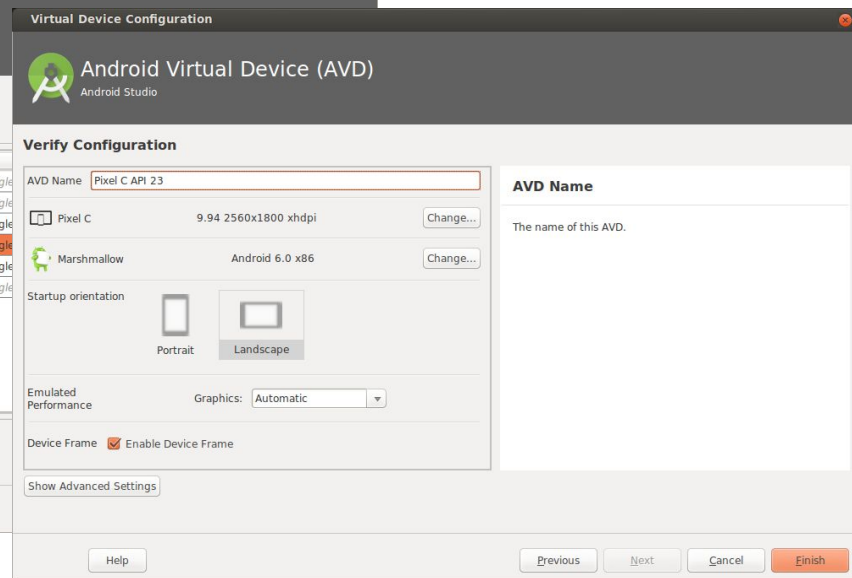
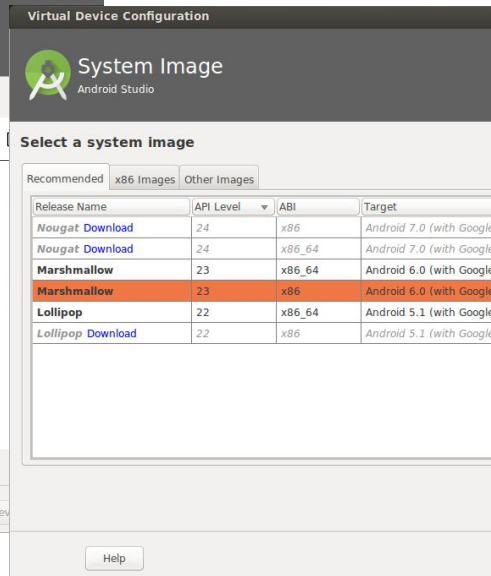
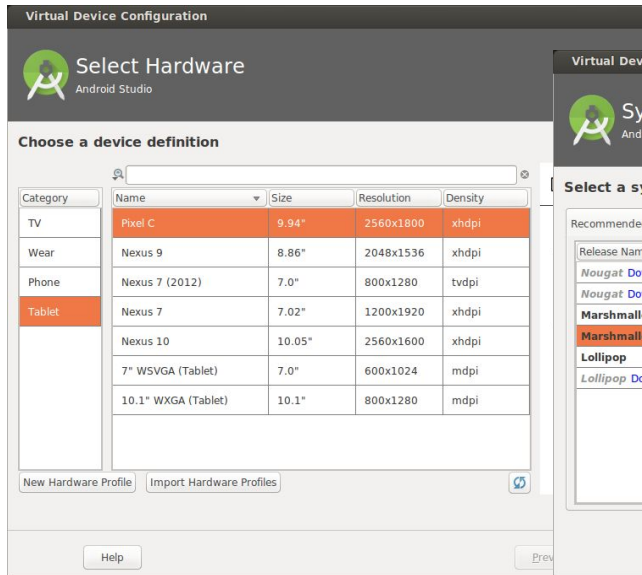


Configure virtual device

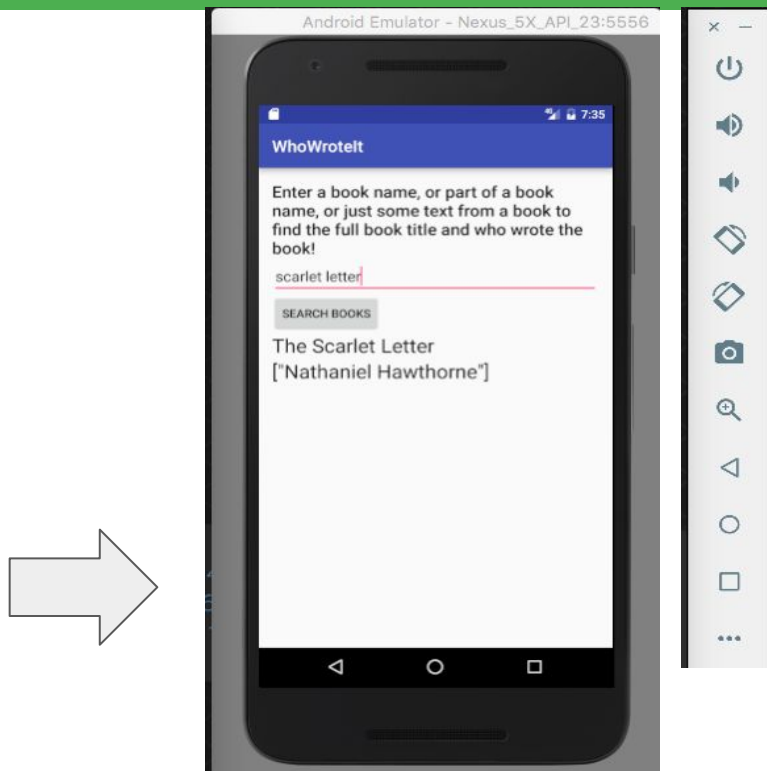
1. Choose hardware

2. Select Android Version

3. Finalize



Run on a virtual device



Run on a physical device

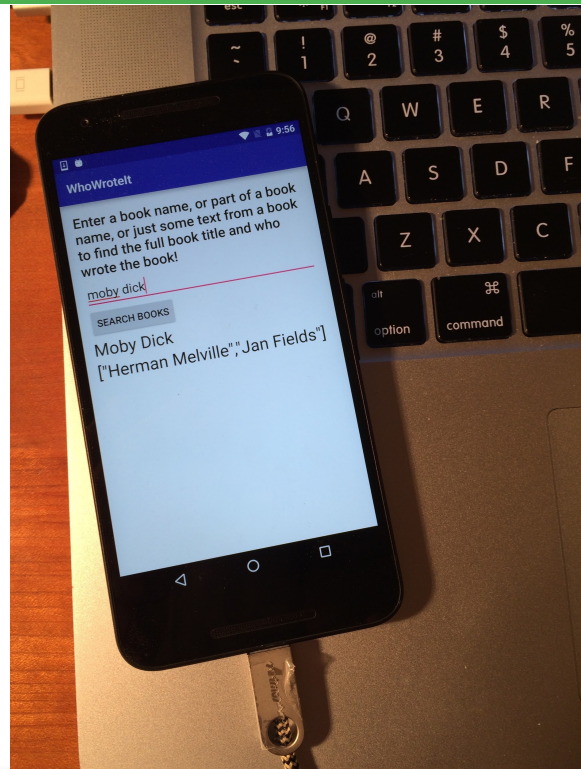
1. Turn on Developer Options:
 - a. **Settings > About phone**
 - b. Tap **Build number** seven times
2. Turn on USB Debugging
 - a. **Settings > Developer Options > USB Debugging**
3. Connect phone to computer with cable

Windows/Linux additional setup:

- [Using Hardware Devices](#)

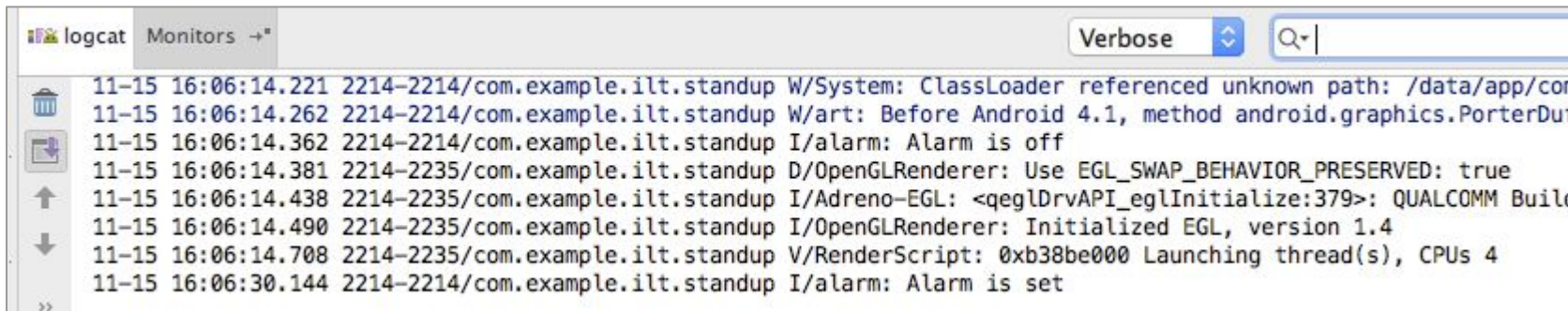
Windows drivers:

- [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.



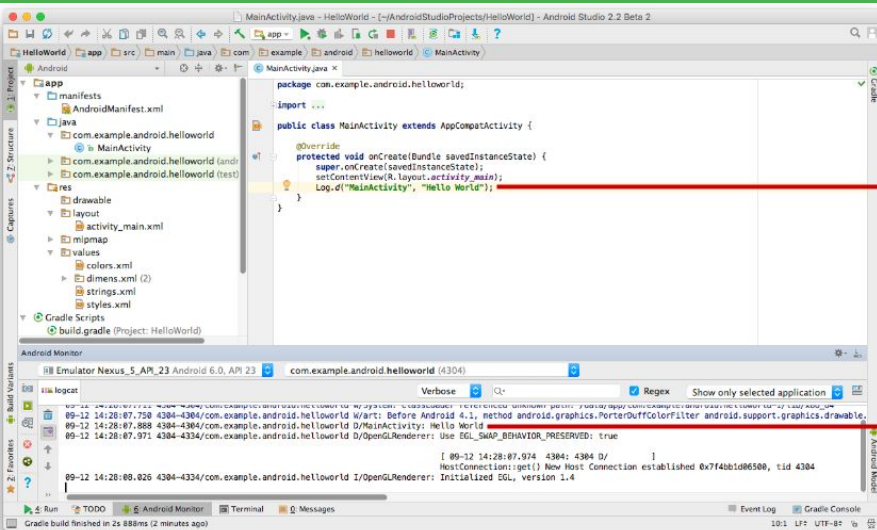
Logging

```
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