Android Studio Setup

Prepare your environment for Android

Android Bundle

Setting up the Development Environment

Android Studio -- Why?

Android Studio is the latest development environment Google recommends for Android. It is based on IntelliJ IDEA which is faster and lighter than **Eclipse**.

When developing apps for Android, we will need **Java SDK**, **Android SDK** as well as the **Android Studio IDE**.

Installing the Java SDK

If you don't have the Java 8 SDK, let's install that first.

1. Go to **http://www.oracle.com/technetwork/java/javase/downloads/index.html**

2. Choose your platform, note that 32-bit and 64-bit versions are both available

3. Download and install the **JDK 8,** Java Development Kit (NOT just the JRE)

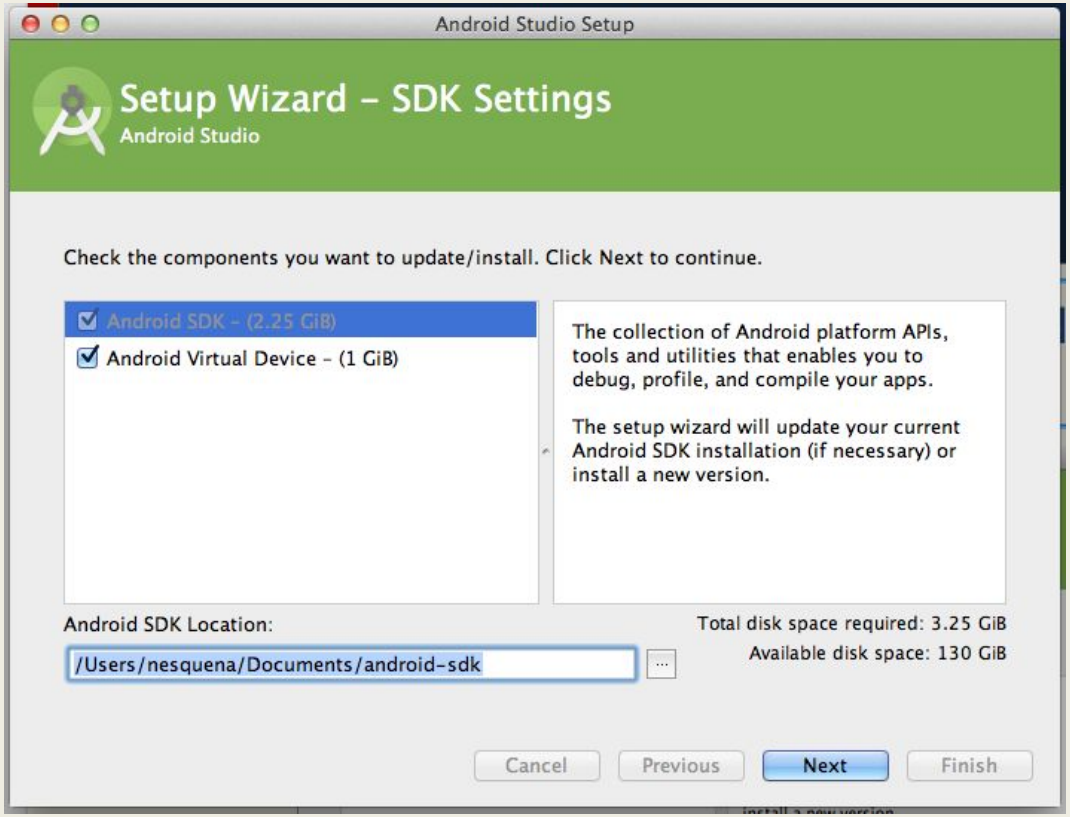
Installing Android Studio

Download Android Studio which also includes an initial setup for Android SDK tools:

1. Download Android Studio http://developer.android.com/sdk/installing/studio.html#download
2. For OS-specific instructions, follow: http://developer.android.com/sdk/installing/index.html?pkg=studio
3. Launch **Android Studio** Application

Installing Android SDK

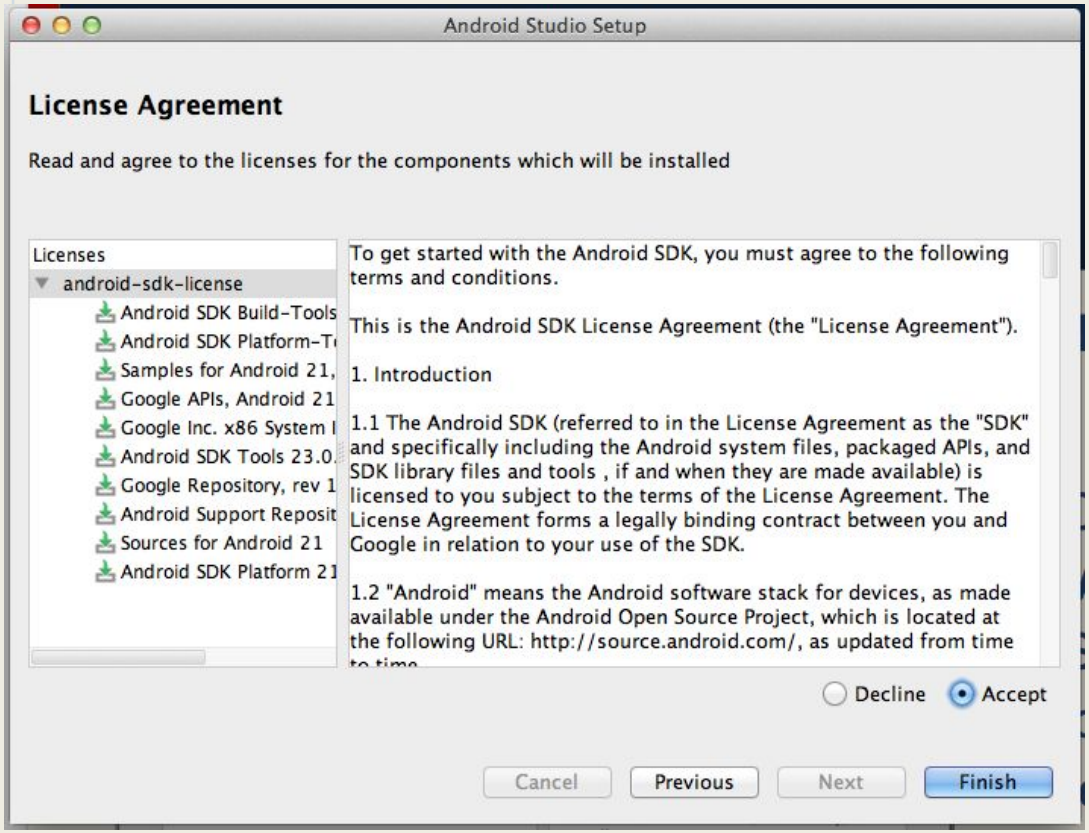
Android Studio should now prompt you to download the **Android SDK** before continuing.



1. Check "**Android Virtual Device**"
2. Select an SDK Location **you will remember**; name folder "**android-sdk**"
3. Click "**Next**" to start install.

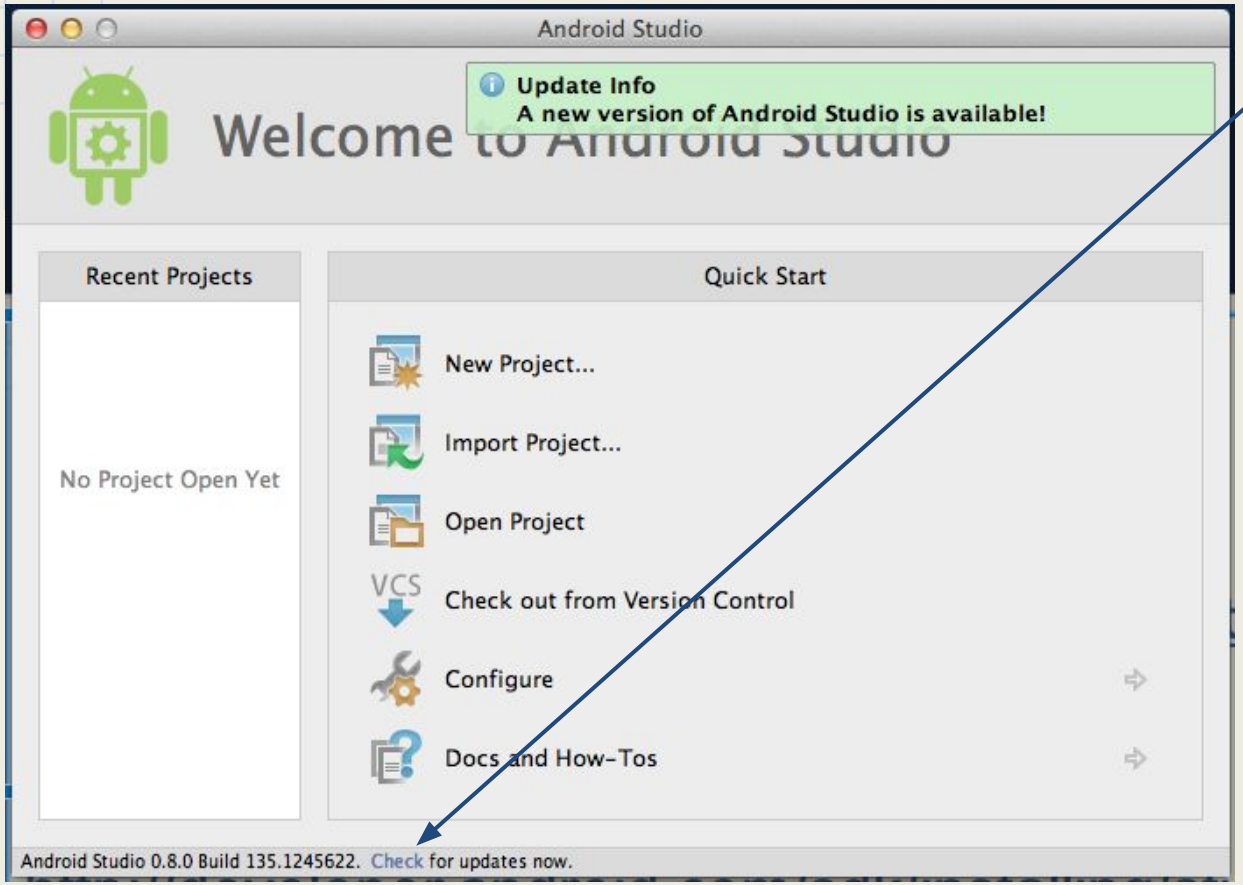
Installing Android SDK

Android Studio should now prompt you to accept the licenses for the SDK.



1. Select the item with the arrow on the left.
2. Select “**Accept**”
3. Click “**Finish**”
4. **Wait** for Setup...then hit “**Finish**” again

Check for Updates

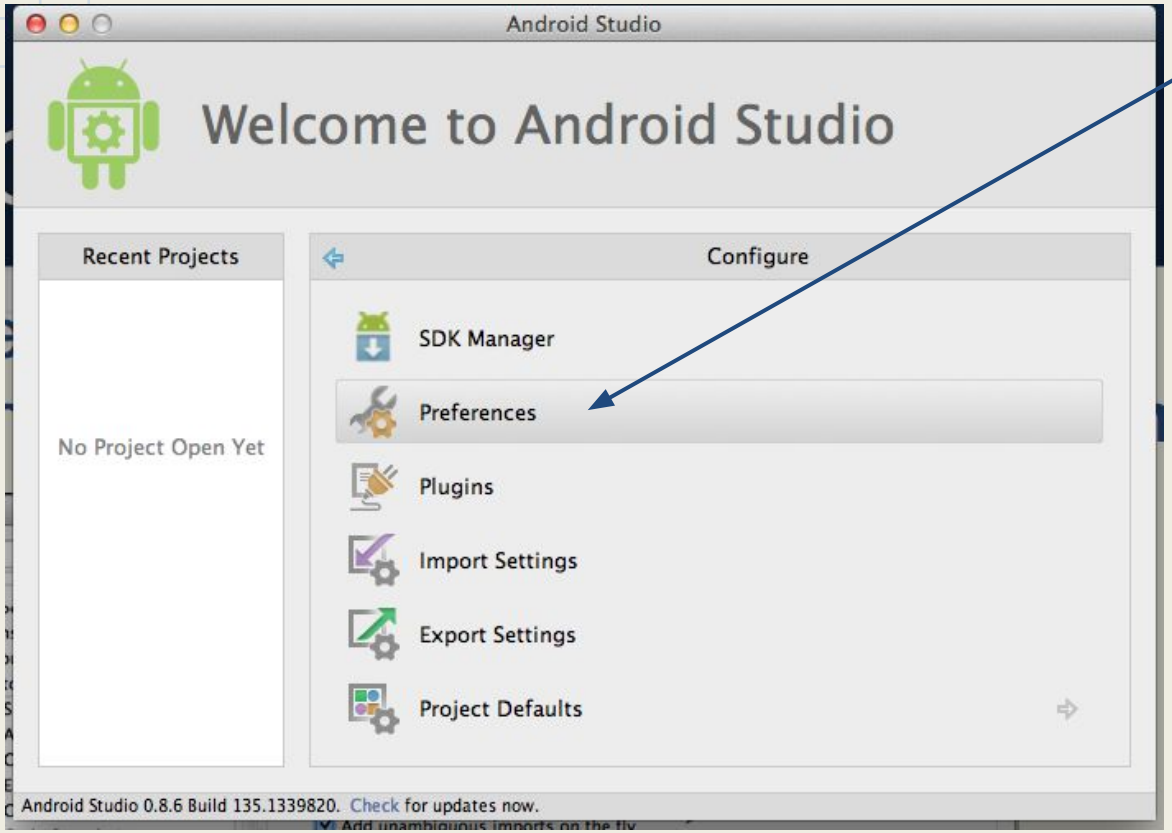


1. Make sure to check for the latest updates.
2. Press "**Update and Restart**" if there is an update.

Configure

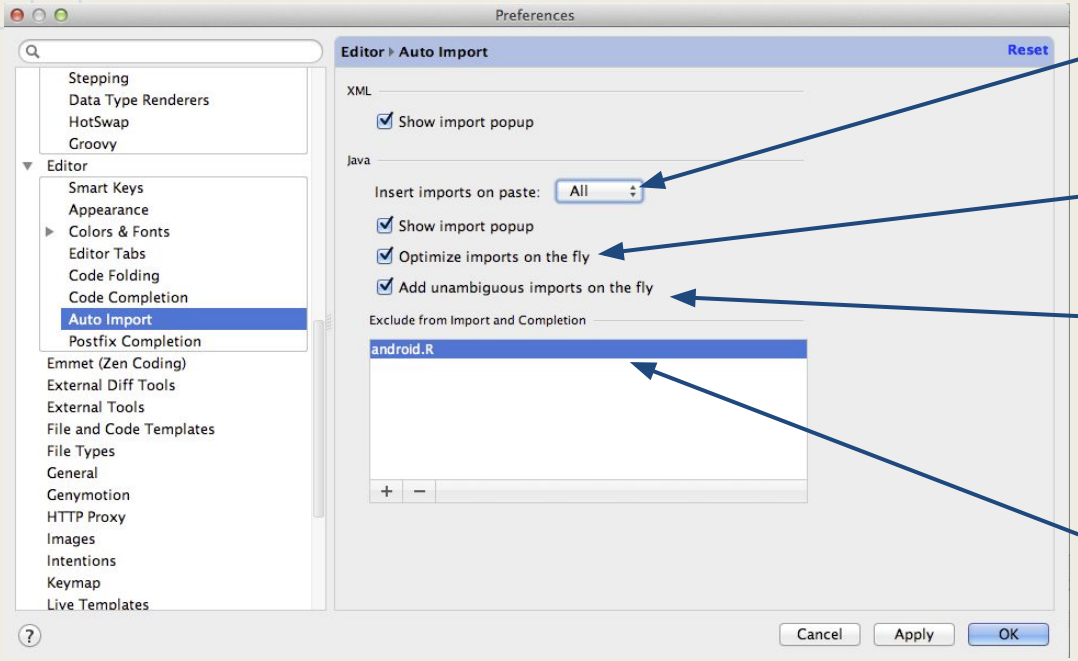
Select "**Configure**" in Menu

Configure Preferences



Select "**Preferences**" in Menu

Configure Imports



1. Set "**Imports on Paste**" to "**All**"
2. Click "**Optimize imports on the fly**"
3. Check "**Add unambiguous imports on the fly**"
4. Exclude **android.R** from auto-imports

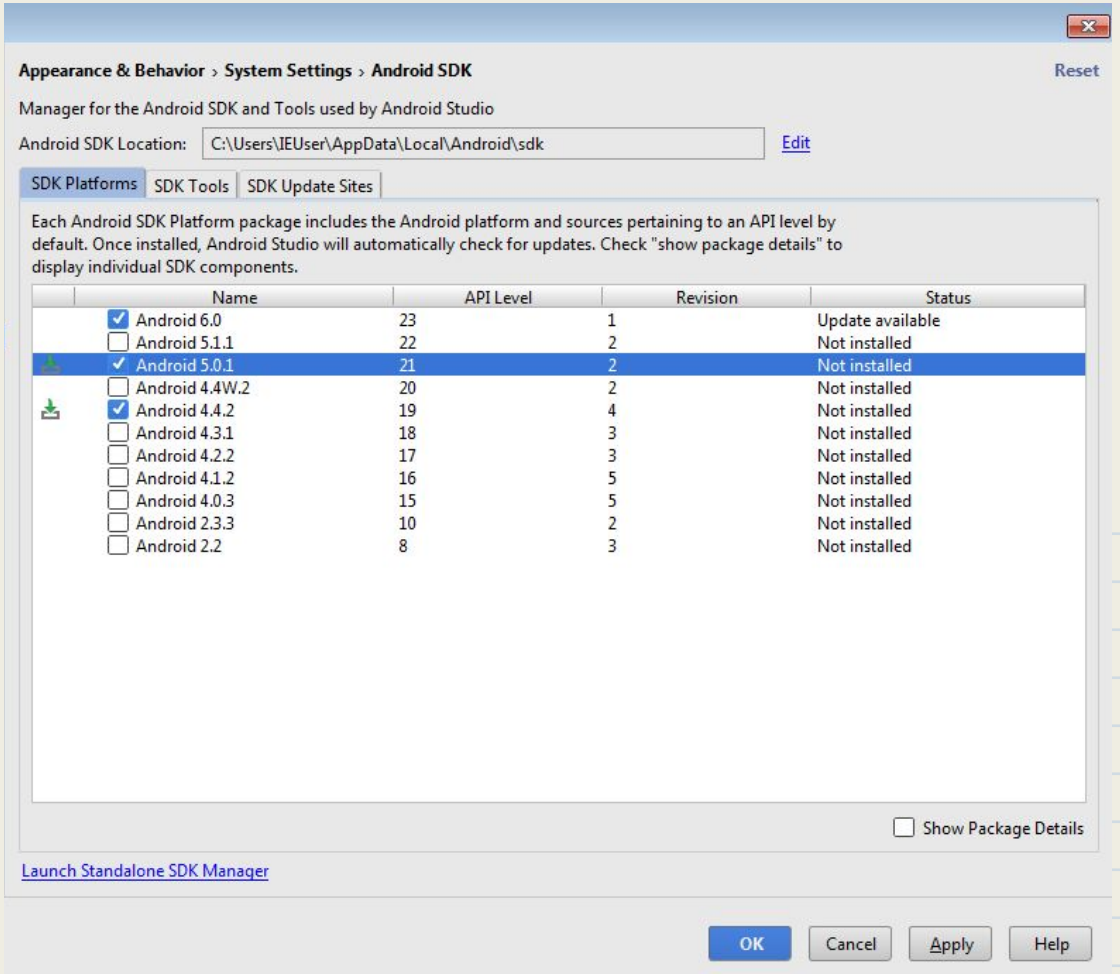
Configure SDKs, Pt 2

Select "**SDK Manager**" in Menu

Android add-ons, 1

Check the following boxes:

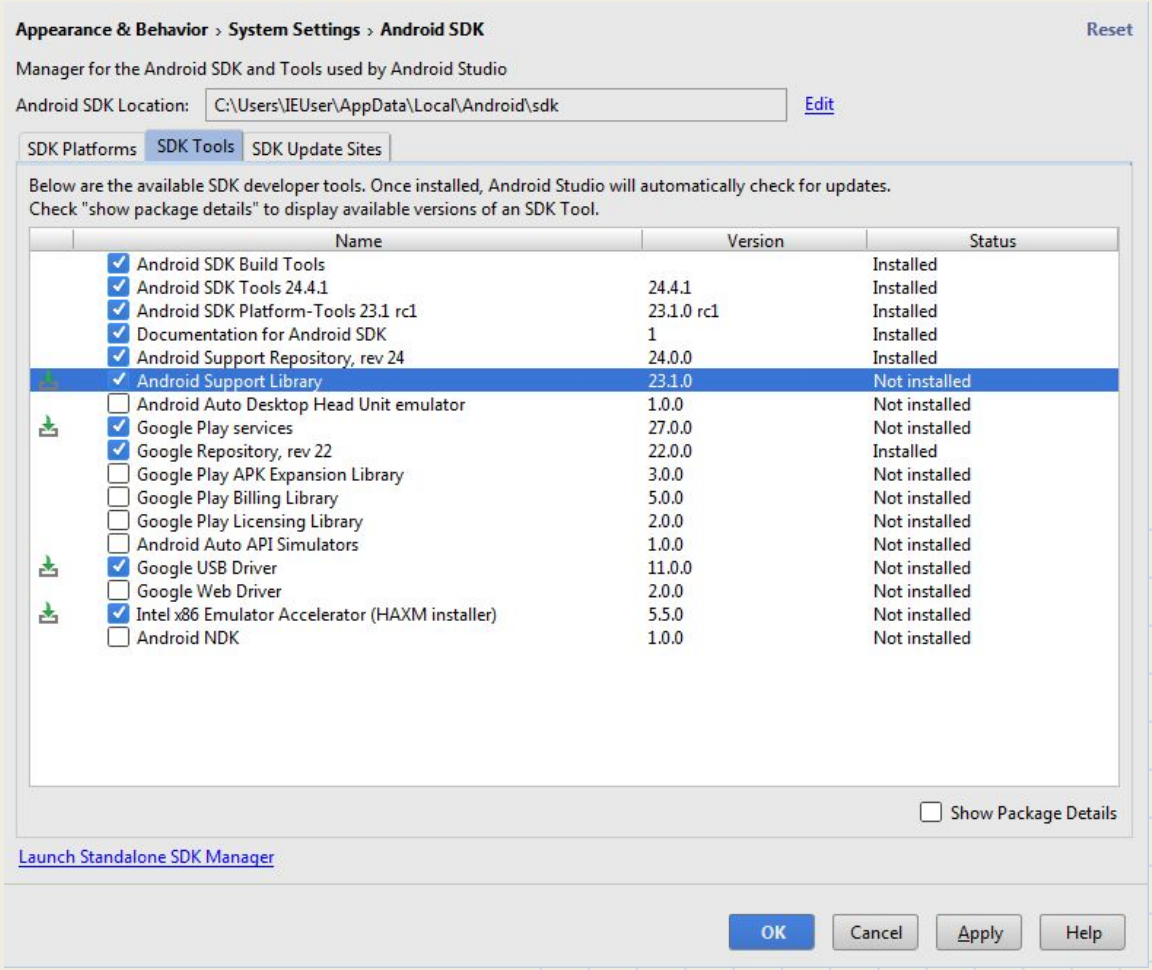
- **Android SDK Tools**
- **Android SDK Platform-tools**
- **Android SDK Build Tools (21)**
- **Android 5.0.1 (API 21)**
  - **Doc, SDK, Samples, Intel x86, APIs**
- **Android 4.4.2 (API 19)**
  - **Doc, SDK, Samples, Intel x86, APIs**
- Extras
  - Android Support Library
  - Google USB Driver (Windows)



Android add-ons, 2

Check the following boxes:

- Android SDK Tools
- Android SDK Platform-tools
- Android 5 (API 21)
  - Doc, SDK, Samples
- **Extras**
  - **Android Support Library**
  - **Google Play services**
  - **Google USB Driver (Windows)**
  - **Intel x86 HAXM**
- **Select 'Install packages...'**



Android emulators

Running Android Apps in Android Studio

Emulator -- Why?

Android Development requires us to **try the apps** we are building while they are being built.

While this can be done by plugging in an Android device, usually it is easier to use an **Emulator**.

An **Emulator** runs the app in a virtual Android on our computer through **Android Studio**.

There are two emulator options: Genymotion or Intel HAXM. We recommend using Genymotion if possible but requires at least 4GB of RAM.

Genymotion

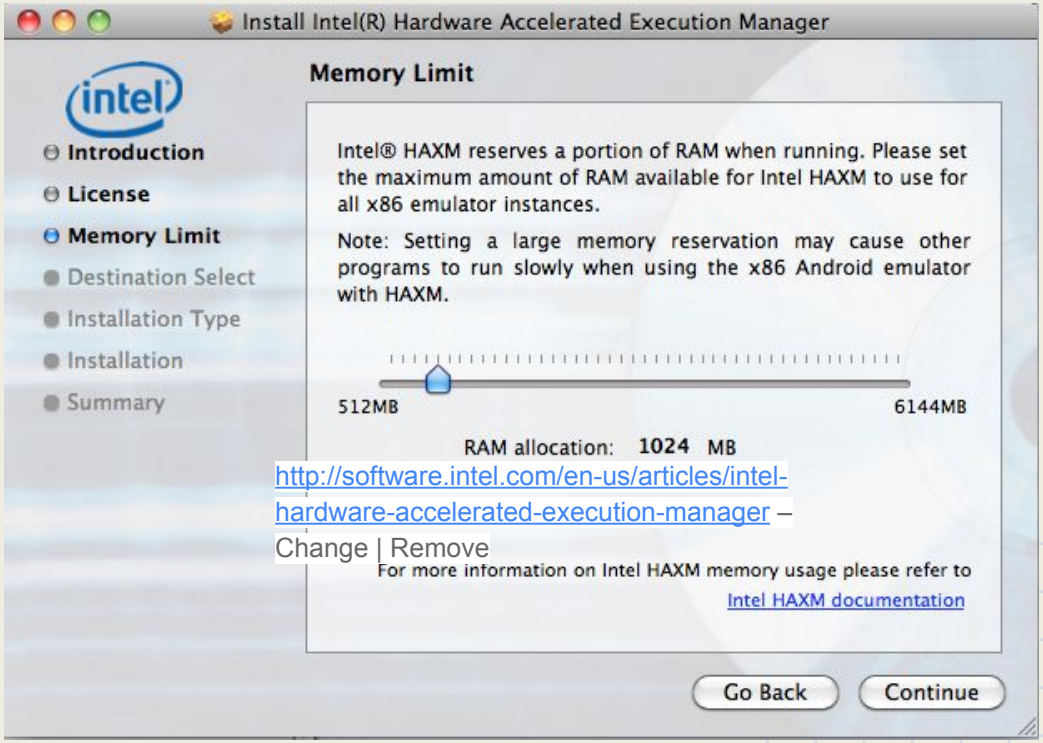
Genymotion requires a few extra steps including registering for an account but is the current recommended Android emulator to use.

http://bit.ly/1IscAhq

On PC's: You must enable Virtualization Technology in the BIOS.

Intel HAXM

1. Download Intel HAXM Emulator installer
2. Run the installer executable and a wizard will be displayed
3. Hit *Next* and select at least 512 MB for memory and continue to hit *Next*
4. Verify the installation completes successfully.



Note: Make sure to download the latest available **hotfix** on the HAXM website for your platform.

Troubleshoot: Intel HAXM

In order for Intel HAXM to work, your computer must support virtualization. In some cases, computers support virtualization, but it is not enabled in the BIOS. In this case, when you try to install Intel HAXM, it will say something like:

"Your hardware supports virtualization, but it is not enabled. Restart your computer, navigate to your BIOS system settings and enable virtualization."

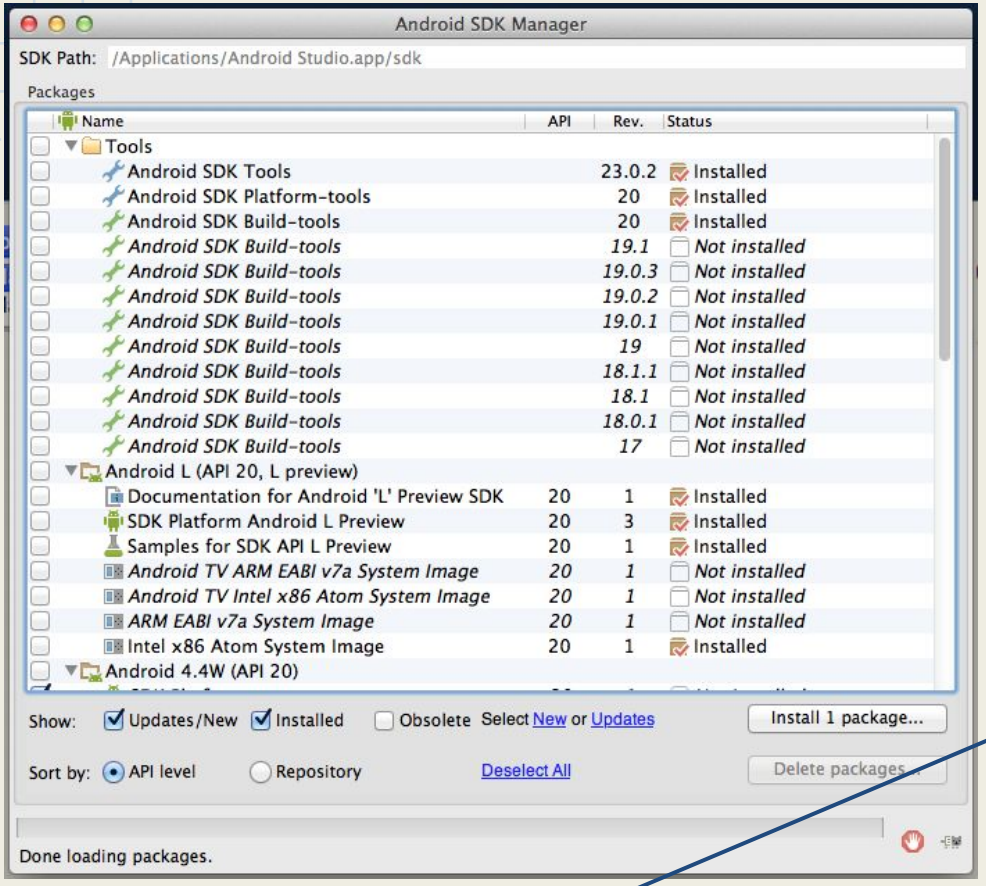
Enable VT Guide (Microsoft Windows): http://www.microsoft.com/windows/virtual-pc/support/configure-bios.aspx

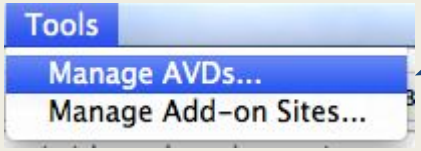
(OSX machines usually have virtualization support already enabled.)

Open up Virtual Devices

Select "**SDK Manager**" in Menu

Open up Virtual Devices



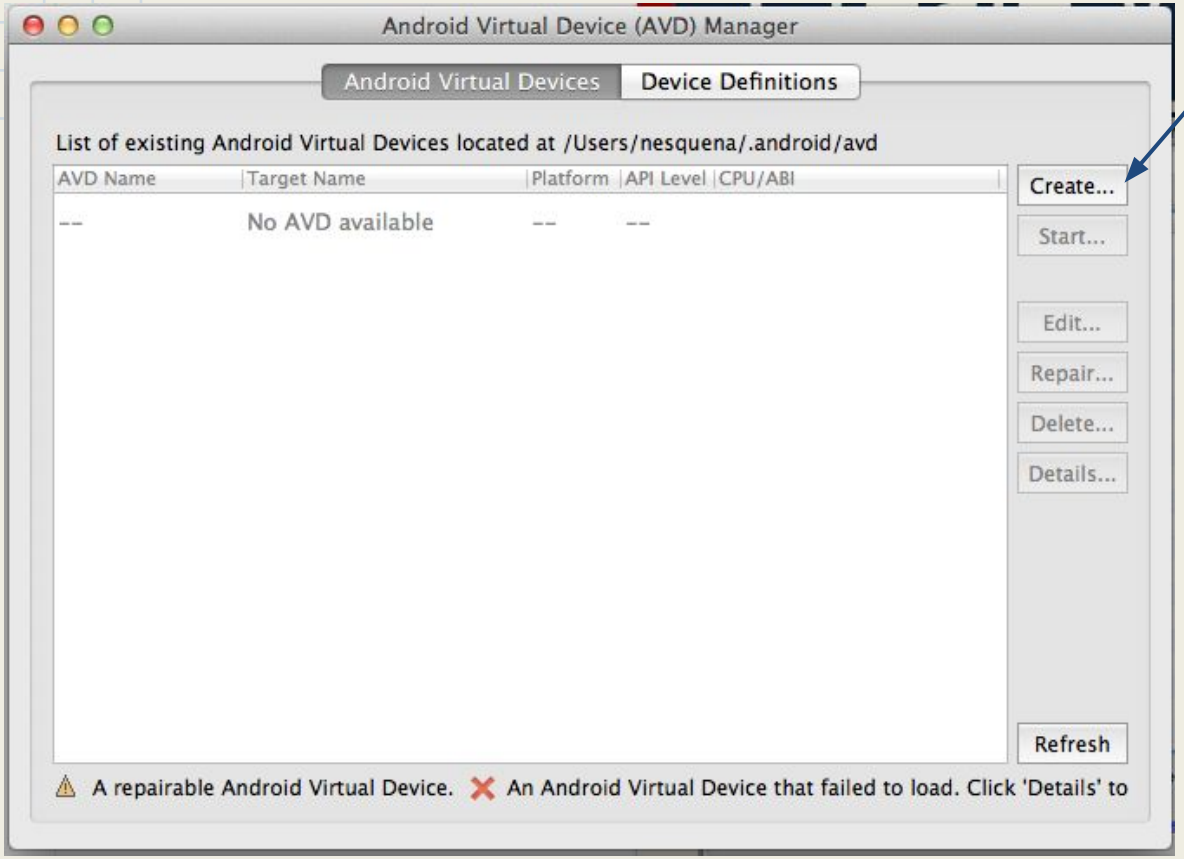


With the SDK Manager Opened…

In the Menu Toolbar:

- Select Tools → **Manage AVDs...**

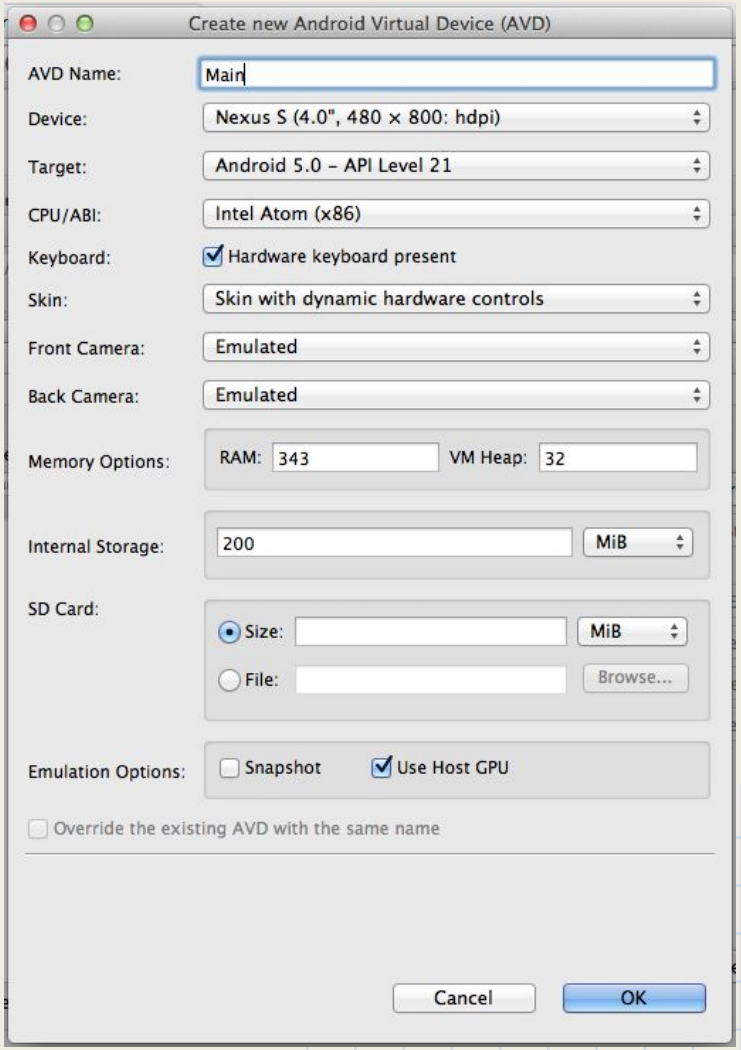
Create Virtual Device



Select **Create…** to add new virtual device

Setup Virtual Devices

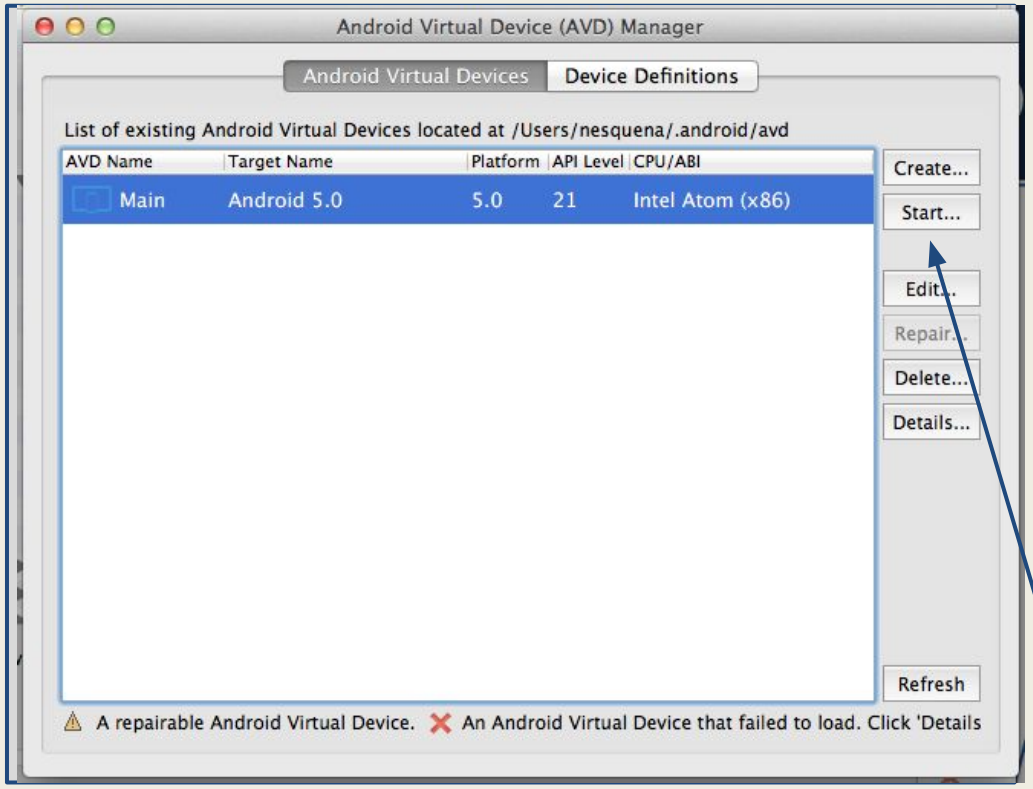
1. AVD Name: **Main**
2. Device: **Nexus S**
3. Target: **Android 5.0**
4. CPU: **Intel Atom (x86)**
5. Skin: **Dynamic**
6. Check **Use Host GPU**
7. Click **OK**



Don't See Intel x86 for CPU?

Be sure you have Intel x86 Atom System Image for the target version and have fully restarted Android Studio after running the Intel x86 HAXM installer.

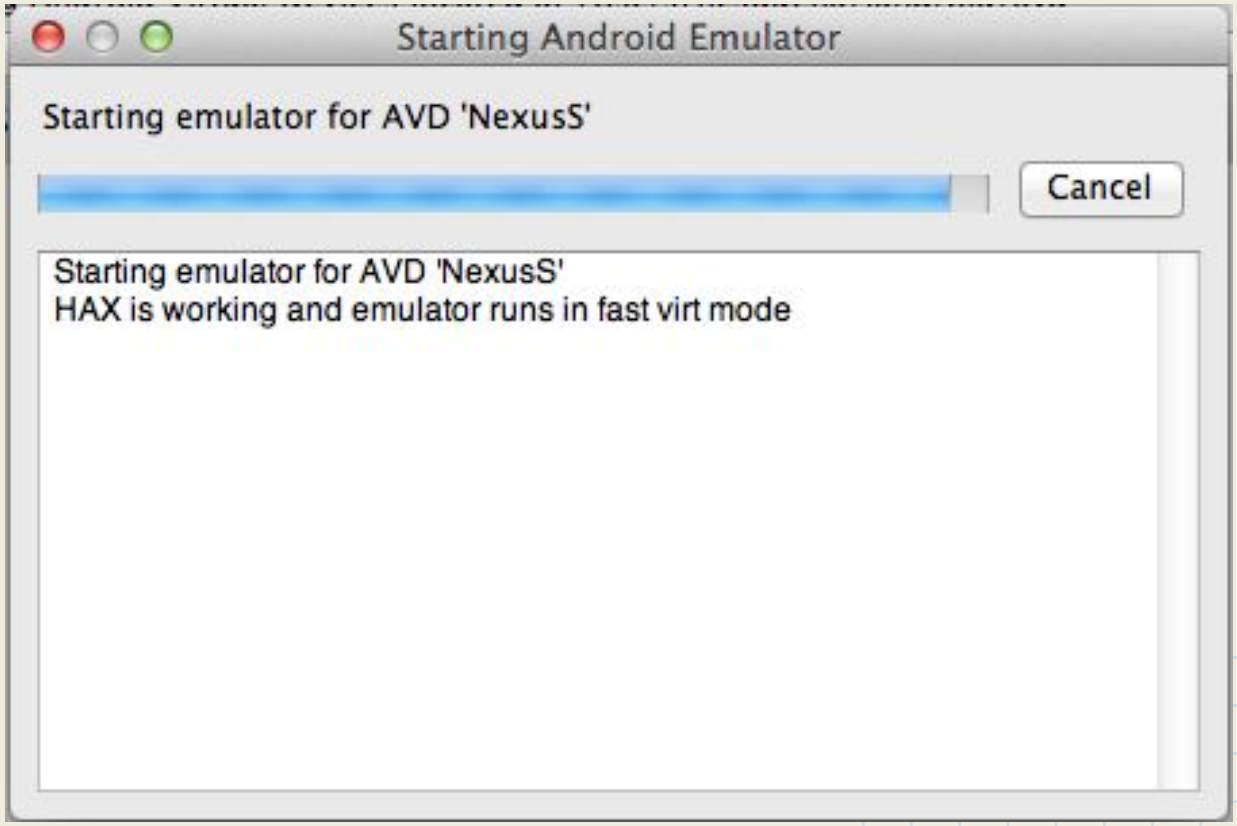
Launch Android Emulator



For the emulator that you create, try clicking on "**Start**..." to then select **Launch**. Starting the emulator for the first time make take a few minutes. **Don't close** the window once it has booted.

Check for "fast virt mode"

If Intel HAXM is set up correctly, as you launch the emulator, you should see the text, "HAX is working and emulator runs in fast virt mode".



Git

Version control with Android Studio

Git -- Why?

When developing software, we should always use **version control** to **manage our code**.

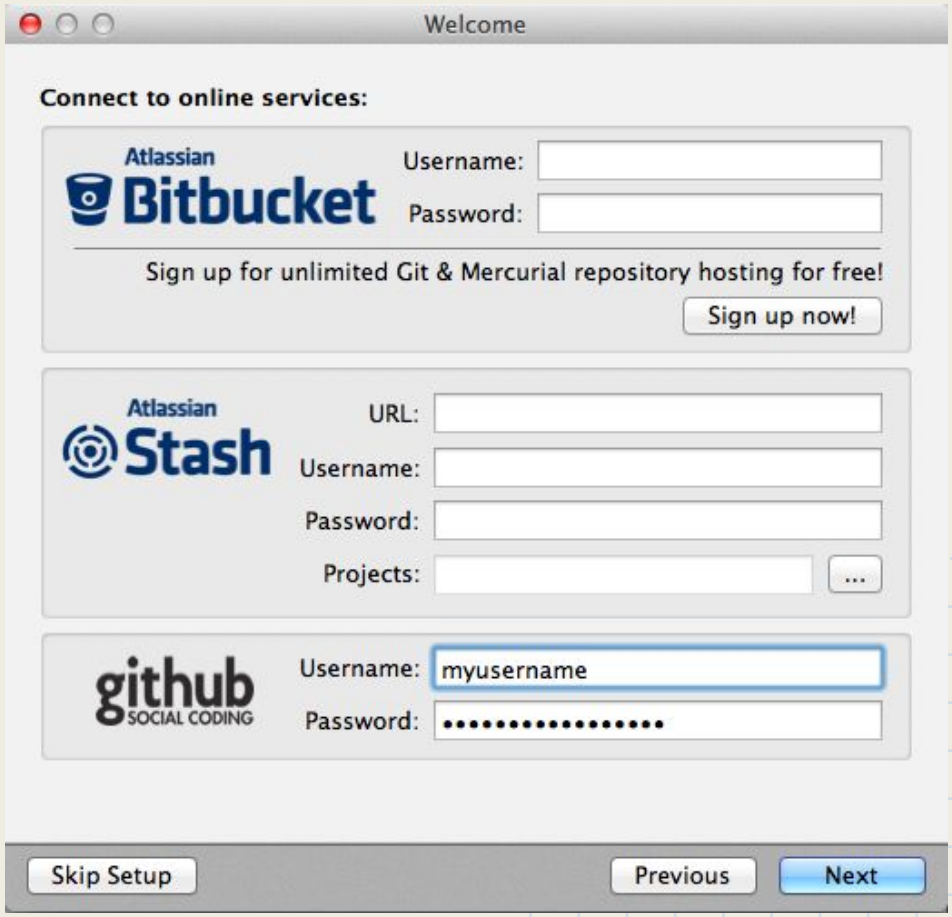
Version control with Git is a way to backup our code, create a version history as we make changes and collaborate on code with other developers.

For our purposes, we will be using **GitHub** to store our code. GitHub is a **free service** for managing our code and saving a **backup** in the cloud.

We will also **download** and **run** our first Android application using Git.

Setup Git Client

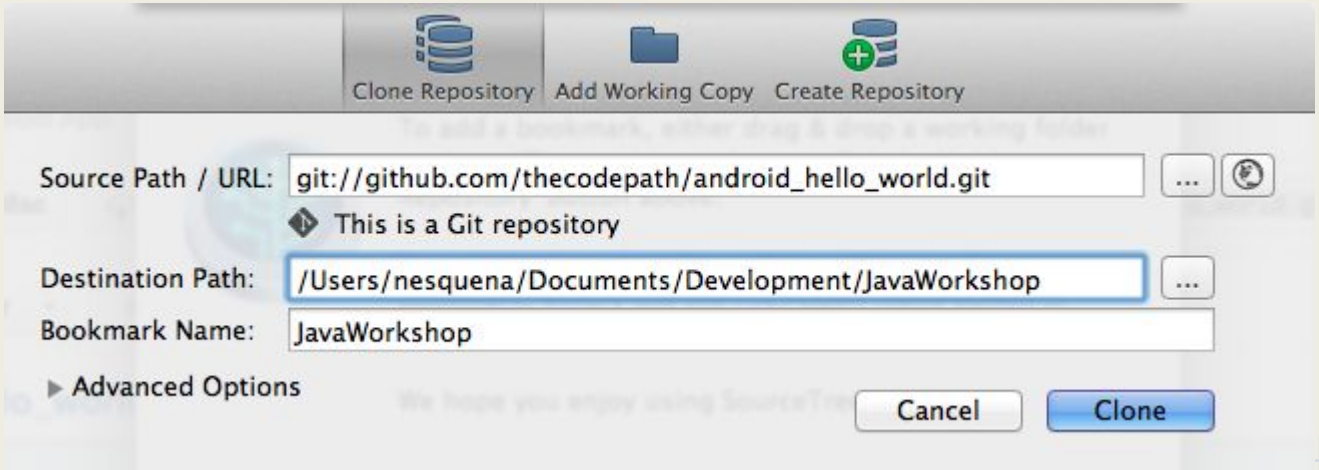
1. Download the **client** from `http://www.sourcetreeapp.com`
2. Install the **sourcetree** client for your platform and **launch** the application.
3. Follow the **setup wizard**, verify your name and email, and enter your **Github** account details (optional).



Note: Or you can just use **git from the command-line**. If you already know git, this is recommended!

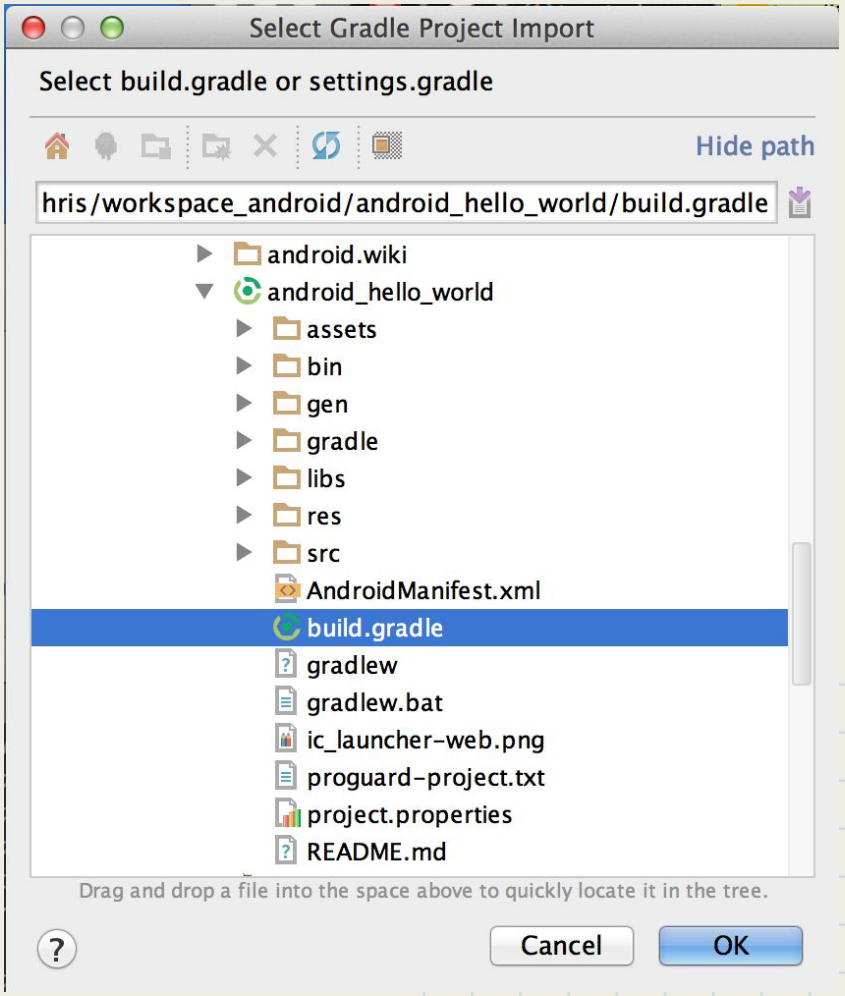
Clone HelloWorldDemo

1. Open **SourceTree**
2. Click File → **New / Clone**
   a. Select **Source Path** as `https://github.com/codepath/android_hello_world.git`
   b. Select a **Destination Path** within Studio Workspace
3. Confirm the **repository** is **listed** within SourceTree UI



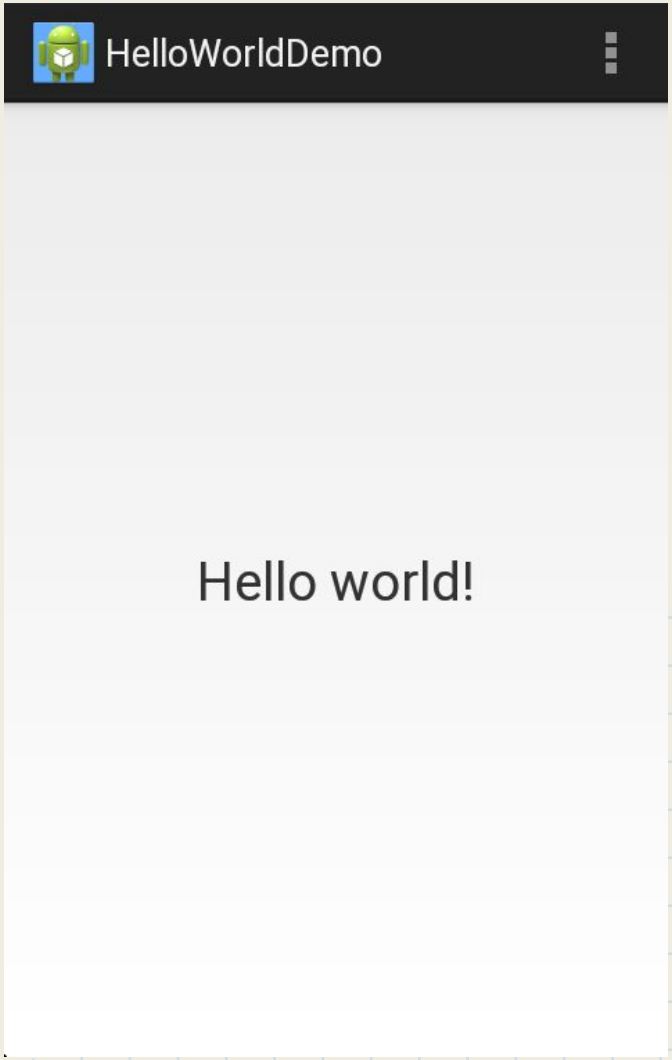
Import HelloWorldDemo

1. In Android Studio, select **Import Project...**
2. In the dialog, select *build.gradle* under the *android_hello_demo* path cloned through Source Tree
3. Click "**Ok**" to Load the Project



Confirm Installation

1. In Android Studio, click on **Run → Run 'android_hello_world'**
2. Emulator should **startup**, and eventually **load** in separate window
3. You should see the application **running** in the **emulator**!



Note: If this doesn't run, try creating a new project instead with `File → New → New Project` and clicking "`Next`" and "`Finish`"

Wrapping Up

Ensuring Setup is Complete

In Review

We have now finished setting up our Android environment including:

- Java and **Android** SDK
- **Android Studio** IDE
- Android **Emulator**
- **Git** Source Control
- Our First Android **Application**