Getting Started with Android Development

By Steven Castellucci (v1.1, January 2015)

You don't always need to be in the PRISM lab to work on your 4443 assignments. Working on your own computer is convenient if the lab is busy, closed, or if you live far from the University. To do so, you must first install software on your computer. The steps below will guide you on downloading, installing, and configuring the needed tools.

Note: The screenshots below were taken on a Windows machine, but the instructions also apply to systems running Linux or OS X.

1. Java Development Kit

If you have already installed the full Java Development Kit (JDK), **and not just** the Java Runtime Environment (JRE), you can skip this step. Otherwise, download and install the JDK at the following URL:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

The JDK version used for this guide is *Java SE 7u40*, but other versions should work equally well.

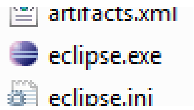
2. Install Eclipse

If you already have Eclipse installed, skip to the next step. Otherwise, download Eclipse IDE for Java Developers from the following URL:

http://www.eclipse.org/downloads/

Eclipse does not require installation. Just extract the files in the zip archive into your home directory. Using your home directory (or a subdirectory therein) ensures that you have the necessary file permissions to read, write, and execute your code.

For convenience, create a shortcut (a.k.a. symbolic link or alias) on the desktop that points to the Eclipse executable. The executable can be found in the Eclipse subdirectory of the extracted bundle.



artifacts.xml
eclipse.exe
eclipse.ini

3. Android SDK Tools and Eclipse Plugin

Download and install the SDK Tools from the following URL (ignore the Android Studio Package):

http://developer.android.com/sdk/index.html#Other

Next, add the ADT plugin to Eclipse:

1. Start Eclipse, then select **Help** > **Install New Software**.
2. Click **Add**, in the top-right corner.
3. In the Add Repository dialog that appears, enter "ADT Plugin" for the *Name* and the following URL for the *Location*:

   https://dl-ssl.google.com/android/eclipse/

   Note: Make sure the update site URL you enter starts with HTTPS.

4. Click **OK**.
5. In the Available Software dialog, select the checkbox next to Developer Tools and click **Next**.
6. In the next window, you'll see a list of the tools to be downloaded. Click **Next**.
7. Read and accept the license agreements, then click **Finish**.

   If you get a security warning saying that the authenticity or validity of the software can't be established, click **OK**.

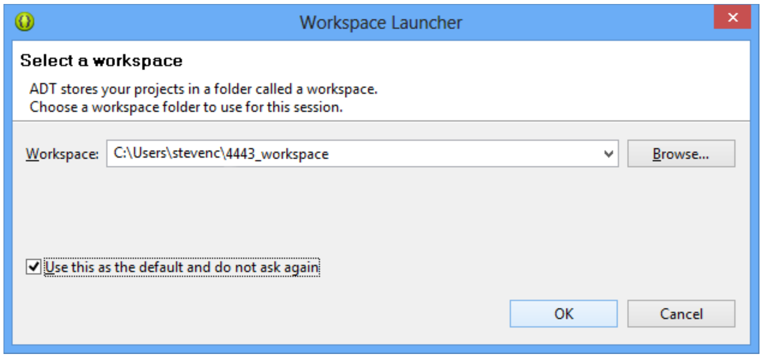
8. When the installation completes, restart Eclipse.

Once Eclipse restarts, you must specify the location of your Android SDK directory:

1. In the "Welcome to Android Development" window that appears, select **Use existing SDKs**.
2. Browse and select the location of the Android SDK directory you recently downloaded and unpacked.
3. Click **Next**.

4. Android IDE

Start Eclipse to run the Android IDE. The dialog below will automatically appear. Change "workspace" to "4443_workspace". This is where your projects will be saved. To skip this step next time, ensure "Use this as the default and do not ask again" is checked. Click `OK` to create the directory (if it does not already exist) and continue. Next, close the "Welcome" tab. Before creating your first Android application, some configuration is required.



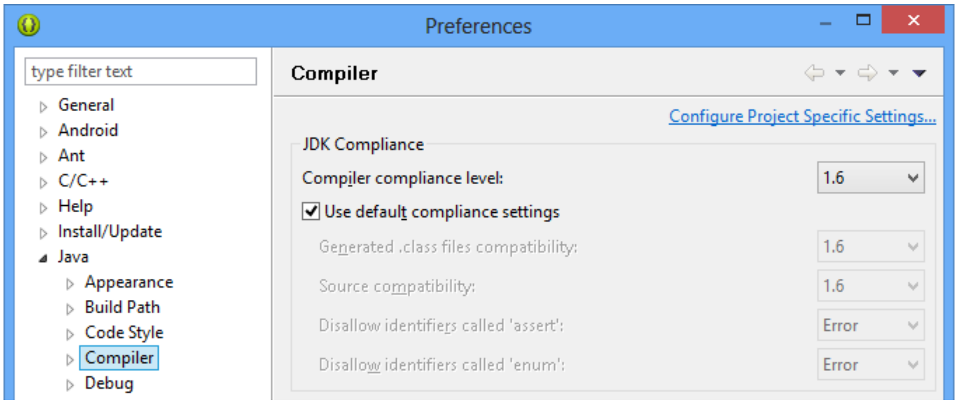
Select a workspace

ADT stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\Users\stevenc\4443_workspace

☑ Use this as the default and do not ask again

Unfortunately, the Android SDK is not yet 100% compatible with Java 1.7. To avoid possible incompatibilities, click on the `Window`menu, then `Preferences`. In the left panel, double-click on `Java`, and then click on`Compiler`. On the right, set “Compiler compliance level” to “1.6”.



Compiler

JDK Compliance

Compiler compliance level: 1.6

☑ Use default compliance settings

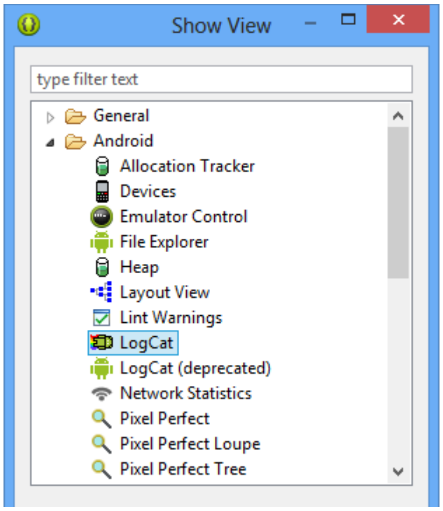
Generated .class files compatibility: 1.6

Source compatibility: 1.6

Disallow identifiers called 'assert': Error

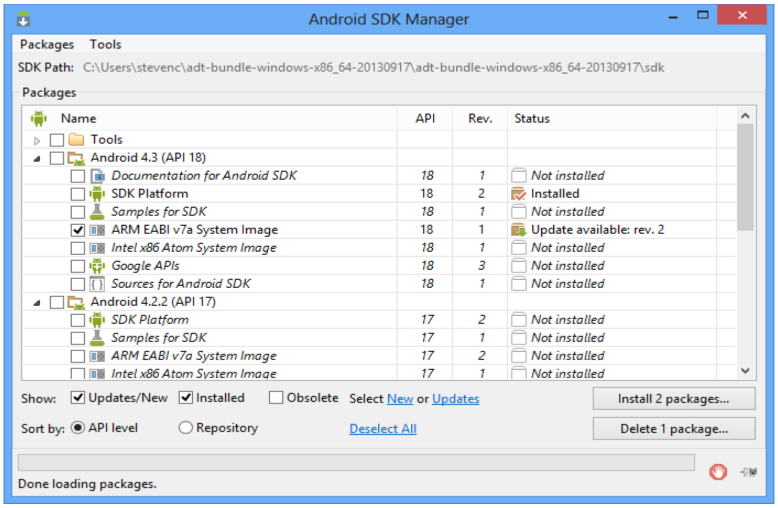
Disallow identifiers called 'enum': Error

Click on the `Window` menu, `Show View`, and `Other…`. Double-click on `Android`, select`Devices` and click `OK`. Repeat the procedure, but this time, select`LogCat`. **Do not** select `LogCat (deprecated)`. Thesenew tabs at the bottom of the IDE window will allow you to view devices and debugging information when running your application.



5. Additional APIs

The SDK bundle comes with the platform and system image of the latest Android API. The platform allows your code to use the features of a specific version of Android. The system image is used by AVDs to emulate the corresponding Android version. Click on the `Window` menu, then `Android SDK Manager`.



Android SDK Manager

SDK Path: C:\Users\stevenc\adt-bundle-windows-x86_64-20130917\adt-bundle-windows-x86_64-20130917\sdk

Packages

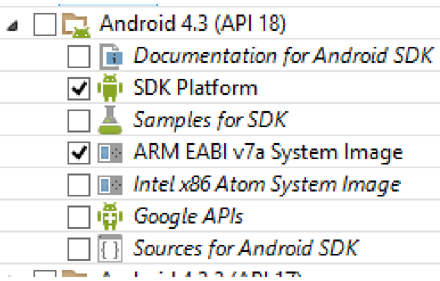
Name	API	Rev.	Status
▷ □ Tools			
◢ □ Android 4.3 (API 18)			
□ *Documentation for Android SDK*	*18*	*1*	*Not installed*
□ SDK Platform	18	2	Installed
□ *Samples for SDK*	*18*	*1*	*Not installed*
☑ ARM EABI v7a System Image	18	1	Update available: rev. 2
□ *Intel x86 Atom System Image*	*18*	*1*	*Not installed*
□ *Google APIs*	*18*	*3*	*Not installed*
□ *Sources for Android SDK*	*18*	*1*	*Not installed*
◢ □ Android 4.2.2 (API 17)			
□ *SDK Platform*	*17*	*2*	*Not installed*
□ *Samples for SDK*	*17*	*1*	*Not installed*
□ *ARM EABI v7a System Image*	*17*	*2*	*Not installed*
□ *Intel x86 Atom System Image*	*17*	*1*	*Not installed*

Show: ☑ Updates/New ☑ Installed □ Obsolete Select New or Updates

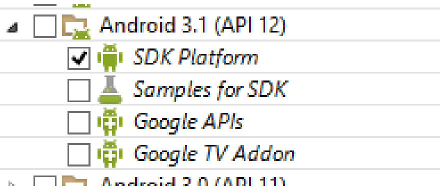
Sort by: ● API level ○ Repository Deselect All

Done loading packages.

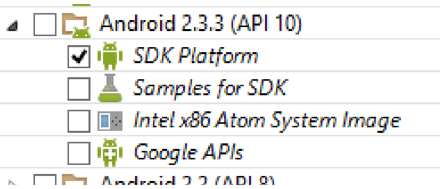
For 4443, we will target the following Android SDK platforms: 5.0 (API 21), 4.3 (API 18), 3.1 (API 12), and 2.3.3 (API 10). In the manager, make the selections illustrated below and click `Install packages`. Accept the license for each package and click `Install`. After installation, it is recommended that you restart Eclipse.



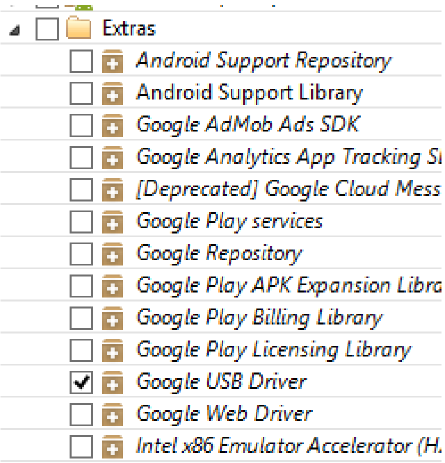
◢ □ Android 4.3 (API 18)
- □ *Documentation for Android SDK*
- ☑ SDK Platform
- □ *Samples for SDK*
- ☑ ARM EABI v7a System Image
- □ *Intel x86 Atom System Image*
- □ *Google APIs*
- □ *Sources for Android SDK*



◢ □ Android 3.1 (API 12)
- ☑ *SDK Platform*
- □ *Samples for SDK*
- □ *Google APIs*
- □ *Google TV Addon*



◢ □ Android 2.3.3 (API 10)
- ☑ *SDK Platform*
- □ *Samples for SDK*
- □ *Intel x86 Atom System Image*
- □ *Google APIs*



◢ □ Extras
- □ *Android Support Repository*
- □ Android Support Library
- □ *Google AdMob Ads SDK*
- □ *Google Analytics App Tracking Si*
- □ *[Deprecated] Google Cloud Mess*
- □ *Google Play services*
- □ *Google Repository*
- □ *Google Play APK Expansion Libra*
- □ *Google Play Billing Library*
- □ *Google Play Licensing Library*
- ☑ *Google USB Driver*
- □ *Google Web Driver*
- □ *Intel x86 Emulator Accelerator (H.*

Note 1: Platforms prior to 4.0 integrated the system image into the platform package.

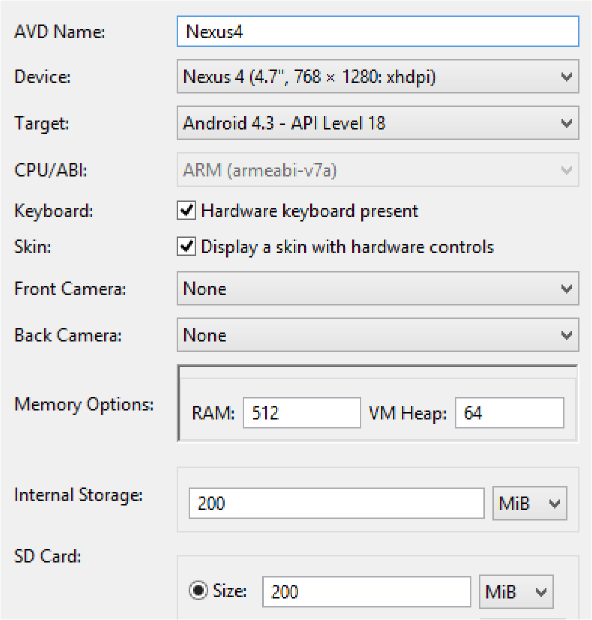
Note 2: The *Google USB Driver* allows you to interface with physical Android devices.

6. Android Virtual Device

You can use an Android Virtual Device (AVD) to run and test your Android applications. To define one, click on the `Window` menu, then `Android Virtual Device Manager`.

Click on `New…`, and use the settings in theimage to the right.In particular, note that the AVD name cannot contain a space.

The device represents the Nexus 4 smartphone, but you could also create an AVD that represents the Nexus 10 tablet. The target API for the device is Android 5.0 (updated from the image). The RAM and SD card sizes are set conservatively. Feel free to increase these values if your computer can spare the resources. Click `OK` to create the AVD.



AVD Name: Nexus4

Device: Nexus 4 (4.7", 768 × 1280: xhdpi)

Target: Android 4.3 - API Level 18

CPU/ABI: ARM (armeabi-v7a)

Keyboard: ☑ Hardware keyboard present

Skin: ☑ Display a skin with hardware controls

Front Camera: None

Back Camera: None

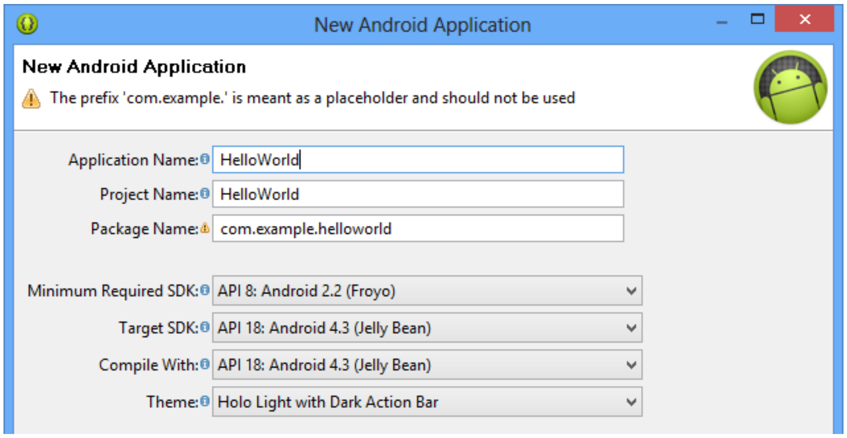
Memory Options: RAM: 512 VM Heap: 64

Internal Storage: 200 MiB

SD Card: ● Size: 200 MiB

7. Hello World

A "Hello World" program will guide you through the process of creating a basic Android application and will test if the IDE and AVD are set up correctly. From the `File` menu, select `New` and `Android Application Project`.Type "HelloWorld" as the application name. For simplicity, click `Next` four times and then `Finish`.



New Android Application

⚠ The prefix 'com.example.' is meant as a placeholder and should not be used

Application Name: HelloWorld

Project Name: HelloWorld

Package Name: com.example.helloworld

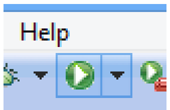
Minimum Required SDK: API 8: Android 2.2 (Froyo)

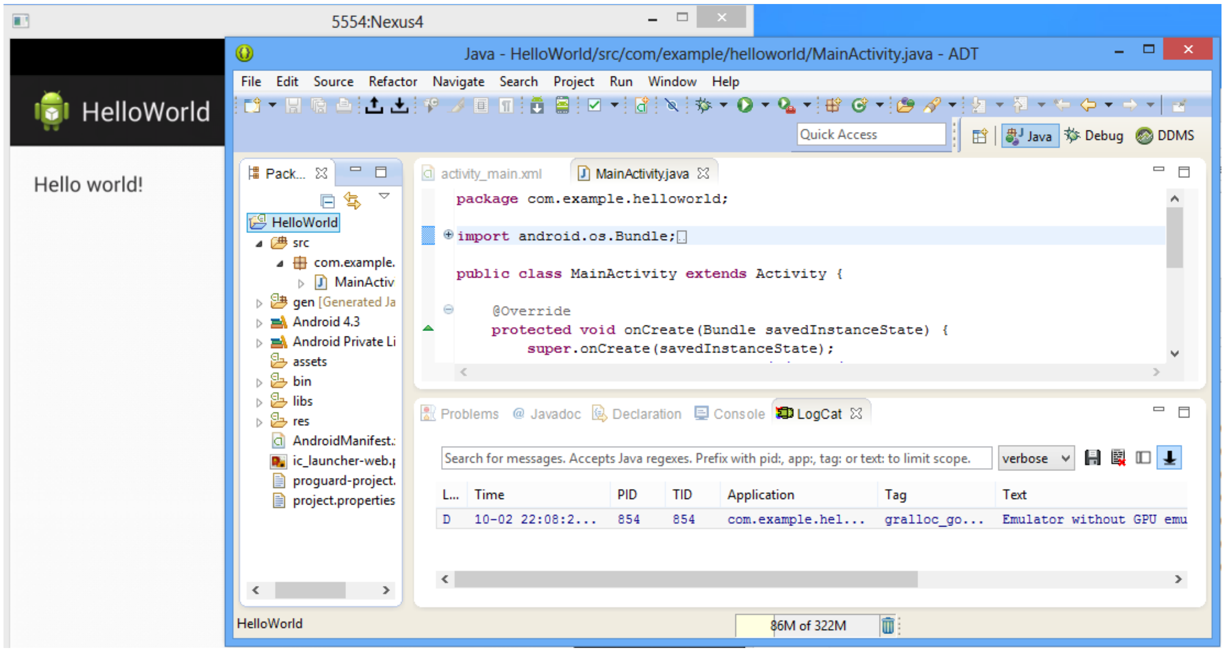
Target SDK: API 18: Android 4.3 (Jelly Bean)

Compile With: API 18: Android 4.3 (Jelly Bean)

Theme: Holo Light with Dark Action Bar

Click the `Run` button on the toolbar, select `Android Application`, and click `OK`. An AVD will launch. After a rather long initialization, your `HelloWorld` application will install and launch.





```
package com.example.helloworld;

import android.os.Bundle;□

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
L...  Time             PID   TID   Application          Tag            Text
D     10-02 22:08:2... 854   854   com.example.hel...   gralloc_go...  Emulator without GPU emu
```

Debugging information from the AVD will appear in the `LogCat` tab at the bottom of the IDE window. For greater access to the AVD, click on the `Window` menu, `Open Perspective`, and `DDMS`. Here, you have access to the AVD's file structure, CPU load, and can simulate incoming phone calls. To return to the previous perspective, click on `Window`, `Open Perspective`, and `Java`.

Note that this application can be installed on an actual device. To do so, copy the `.apk` file found in the project's `bin` directory to the actual device. Running the `.apk` file on the device will install it.