## APPLICATION NOTE

# How to build your first Android Application in Windows

3/30/2012 Created by: Micah Zastrow

## Abstract

This application note is designed to teach the reader how to setup the Android Development Environment by using Eclipse, and how to create the readers first Android Application. It first shows how to install and setup all of the software that is needed. It then shows how to create an Android project and how to enter sample code to create an Android application. Lastly, it shows how to either use a virtual device, or connect an android device to the computer and run the application on either of these devices.

## Keywords

**Eclipse:** Eclipse is a platform that has been designed from the ground up for building integrated web and application development tooling.

**IDE:** is a software application that provides comprehensive facilities to computer programmers for software development.

**Software Development Kit (SDK):** is typically a set of software development tools that allows for the creation of applications for a certain software package

**Android Development Tools (ADT):** A plugin for eclipse that extends the Eclipse IDE by providing more tools to develop Android Applications.

Java SE Development Kit (JDK): A popular Java SDK that is used to program Android applications.

<u>Android Virtual Device (AVD)</u>: An Android emulator that allows you to simulate how the application will run on an actual Android device.

### Introduction

Android applications, like most mobile phone applications, are developed in a host-target development environment. In other words, you develop your application on a host computer, and download it to a target mobile phone for testing and ultimate use. To write your own Android mobile phone applications, you'll first need to collect the required tools and set up an appropriate development environment on your PC. The Android SDK supports several different development environments. For this application note, we will focus on using Eclipse, because it is the best integrated with the SDK.

## Objective

This application note provides a practical introduction to developing applications for. It explores the framework for constructing an application, and the tools for developing, testing, and publishing software for the platform.

# System Requirements

The sections below describe the system and software requirements for developing Android applications using the Android SDK. <u>http://developer.android.com/sdk/requirements.html</u>

### **Supported Operating Systems**

• Windows XP (32-bit), Vista (32- or 64-bit), or Windows 7 (32- or 64-bit)

### Supported Development Environments Eclipse IDE

- Eclipse 3.6.2 (Helios) or greater
- Eclipse JDT plugin (included in most Eclipse IDE packages)
- If you need to install or update Eclipse, you can download it from http://www.eclipse.org/downloads/.
- Several types of Eclipse packages are available for each platform. For developing Android applications, we recommend that you install one of these packages:
  - Eclipse IDE for Java Developers
  - Eclipse Classic
  - Eclipse IDE for Java EE Developers
  - JDK 6 (JRE alone is not sufficient)
  - Android Development Tools plugin (recommended)

#### Hardware requirements

The Android SDK requires disk storage for all of the components that you choose to install. The table below provides a rough idea of the disk-space requirements to expect, based on the components that you plan to use.

Component type	Approximate size	Comments
SDK Tools	35 MB	Required.
SDK Platform-tools	6 MB	Required.
Android platform (each)	150 MB	At least one platform is required.
SDK Add-on (each)	100 MB	Optional.
USB Driver for Windows	10 MB	Optional. For Windows only.
Samples (per platform)	10M	Optional.
Offline documentation	250 MB	Optional.

# Installation

**Eclipse:** Eclipse is the preferred development environment for Android. You can download eclipse at <u>http://www.eclipse.org/downloads/</u>. It is recommended that you install either classic or the Java Version of eclipse.



**JDK:** Download the latest JDK by going here: http://www.oracle.com/technetwork/java/javase/downloads/index.html

**Android SDK:** Go to <u>http://developer.android.com/sdk/index.html</u> and click on the Windows installer.

ackages Tools				
DK Path: C:\Program Files (x86)\Android\android-sdk				
Packages				
👾 Name	API	Rev.	Status	
a 🔽 🧰 Tools				
V X Android SDK Tools		16	🖊 Update available: rev. 17	
📝 🙀 Android SDK Platform-tools		10	🖊 Update available: rev. 11	
a 🔄 🚔 Android 4.0.3 (API 15)				
🔽 🚞 Documentation for Android SDK	15	1	🖊 Update available: rev. 2	
👿 👘 SDK Platform	15	2	🐥 Update available: rev. 3	
🔽 🌄 Samples for SDK	15	1	🐥 Update available: rev. 2	
📄 👾 ARM EABI v7a System Image	15	1	🎒 Installed	
📄 🖏 Google APIs by Google Inc.	15	1	🎒 Installed	
Sources for Android SDK	15	1	🎒 Installed	
Image: Provide Android 4.0 (API 14)				
Marcel Android 3.2 (API 13)				
Image: Marcel Android 3.1 (API 12)				
Android 3.0 (API 11)				
ihow: 🗹 Updates/New 📝 Installed 🛛 🕅 Obsolete Selec	t <u>New</u> or <u>Up</u>	<u>dates</u>	Install 5 pack	ages
ort by:  API level C Repository Dese	lect All		Delete 5 pack	ages

#### **Eclipse ADT Plugin:**

- 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*: <u>https://dl-ssl.google.com/android/eclipse</u>
- 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**.
- 8. When the installation completes, restart Eclipse.

Installed Software Available Software		
type filter text		Install
Name	Version	
<ul> <li>I Eclipse Project Update Site</li> <li>EPP Usage Data Collector Update Site</li> <li>Ganymede</li> <li>Intps://dl-ssl.google.com/android/eclipse/</li> <li>Developer Tools</li> <li>Android Development Tools</li> <li>Android Editors</li> <li>Mylyn for Eclipse 3.4</li> <li>Web Tools (WTP) Update Site</li> </ul>	0.8.0.v200809220836-110569 0.8.0.v200809220836-110569	Properties Add Site Manage Sites. Refresh
<		
Include items that have already been installed		
pen the <u>'Automatic Updates'</u> preference page to set up an automatic	update schedule.	Close

#### **Configuring the ADT Plugin**

After you've successfully downloaded the ADT as described above, the next step is to modify your ADT preferences in Eclipse to point to the Android SDK directory:

- 1. Select **Window** > **Preferences...** to open the Preferences panel
- 2. Select **Android** from the left panel.
- 3. For the *SDK Location* in the main panel, click **Browse...** and locate your downloaded SDK directory.
- 4. Click **Apply**, then **OK**.

type filter text	Android			$\langle \phi \star \phi \star \star \star$
<ul> <li>General</li> <li>Android</li> <li>Ant</li> <li>Data Management</li> <li>Help</li> <li>Install/Update</li> <li>Java</li> <li>JavaScript</li> <li>JPA</li> <li>Plug-in Development</li> <li>Remote Systems</li> <li>Run/Debug</li> <li>Server</li> <li>Service Policies</li> <li>Tasks</li> <li>Team</li> <li>Usage Data Collector Validation</li> <li>Web</li> <li>Web Services</li> <li>XDoclet</li> <li>XML</li> </ul>	Android Prefere SDK Location:	ences C:\eclipse\android-sdk-v	vindows-1.0_r1	Browse
0			ОК	Cancel

#### Adding SDK Components.

- 1. Click on Window->Android SDK Manager.
- 2. This will allow you to choose the Android platform versions, add-ons tools and other components. Choose the version of Android that you would like you application to work on.
- 3. Click Install Selected and wait for the components to down. When the download is finished, verify and accept the new components.

#### **Creating an Android Application**

- 1. In Eclipse go to File->New->Project...
- 2. Select an Android Project from the Android Folder and press Next.
- 3. Fill in the details of your Android application.
  - a. Project Name: The project name and folder that Eclipse will store the

project files

b. Build Target: The version of the Android SDK that will be used when you build your program. Select a platform that is equal to or lower than the target chosen for the AVD.

c. Application Name: This is the name of the application.

d. Package Name: The namespace that all of the source code will reside under.

e. Create Activity: The name for that class stub that is generated by the plugin.

- 4. The values that are used in this example are:
  - a. Project Name: HelloAndroid
  - b. Build Target: 2.1
  - c. Application Name: Hello, Android
  - d. Package Name: com.example.helloandroid
  - e. Create Activity: HelloAndroid
- 5. Click on Finish.

New Android Project	
New Android Project Creates a new Android Project resource	e.
Project name: Contents © Create new project in workspace © Create project from existing source V Use default location	2
Location: C:/Users/Dave/workspace	e Browse
Properties Package name: Activity name: Application name:	
0	<u>F</u> inish Cancel

😂 New Android Proje	ct						
New Android Proj Creates a new Andro	jec id P	t roject resource.					
Project name: Hello	oAn	droid					
Contents							
<ul> <li>Create new proje</li> </ul>	ct ir	n workspace					
Create project fro	m	existing source					
Use default locati	ion	-					
Location: C:/User	s/sr	nain/workspace/HelloAndroid	H (	Brov	vse		
Create project fro	m	existing sample					
Samples: This targ	et k	as no samples. Please select a	nothe	r target.	-		
Build Target							
Target Name		Vendor	Р	latform	API		
Android 1.5		Android Open Source Project		.5	3		
Android 1.6		Android Open Source Project	t 1	.6	4		
Android 2.1-u		Android Open Source Project		.1-up	7		
Android 2.2		Android Open Source Projec	t 2	.2	8		
Android 2.3		Android Open Source Projec	t 2	.3	9		
Standard Android p	olatf	orm 2.1-update1					
Properties							
Application name:	He	ello, Android					
Package name:	co	om.example.helloandroid					
Create Activity:	He	elloAndroid	oAndroid				
Min SDK Version:	7						
? < <u>B</u> ack		<u>N</u> ext > <u>Finis</u>	h		Cancel		

## Writing the Code

Your Android project is now ready. It should be visible in the Package Explorer on the left. Open the HelloAndroid.java file, located inside *HelloAndroid* > *src* > *com.example.helloandroid*). It should look like this:

package com.example.helloandroid;

import android.app.Activity; import android.os.Bundle;

```
public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

Take a look at the revised code below and then make the same changes to your HelloAndroid class. The bold items are lines that have been added.

package com.example.helloandroid;

import android.app.Activity; import android.os.Bundle; import android.widget.TextView;

```
public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
        tv.setText("Hello, Android");
        setContentView(tv);
    }
}
```

# Run the Application

### Adding an AVD

- 1. In Eclipse go to Window->AVD Manager->New.
- 2. Type in the Name of the AVD and choose a Target. The target is the version of the Android SDK that you would like to run on the emulator.
- 3. Click Create AVD.
- 4. Select **Run > Run**.
- 5. Select "Android Application".

#### Using an Android Device (Optional)

If you want to use your own Android Device, then you can follow these steps to setup eclipse to use your device.

- 1. Select Run > Run Configurations.
- 2. Select your project
- 3. Click on the Target tab
- 4. Make sure "Manual" is checked under Deployment Target Selection Mode
- 5. Click run
- 6. When the Android Device Chooser opens select your android device and click OK.

) 🗊 🗶 🖻 🏇 🔹	Name:	HelloAndroid						
ype filter text	Android Target Common							
<ul> <li>Android Application</li> <li>HelloAndroid</li> <li>team8</li> <li>Tictactoe (I)</li> <li>Tipcalculator</li> <li>Ju Android JUnit Test</li> <li>Backerry Android App</li> <li>C/C++ Application</li> <li>C/C++ Remote Applicat</li> <li>Java Applet</li> <li>Java Applet</li> <li>Java Application</li> <li>Ju JUnit</li> </ul>	Deployment Target Selection Mode   Manual   Automatic  Select a preferred Android Virtual Device for deployment:							
		AVD Name	Target Name	Platform	API Level	CPU/ABI	Details	
		🗌 droid	Android 2.3.3	2.3.3	10	ARM (arm	Start	
Launch Group	Emula	tor launch para	imeters:				Refresh Manager	
	Network Speed: Full -							
	Wip Dis Additio	be User Data able Boot Anim onal Emulator (	ation Command Line Optic	ins				

	Serial Number		AVD Nar	ne	Target	Debu	g Stat	te	
	333390F11ADF00EC		N/A	N/A		Yes	On	Online	
) Lau	inch a new Andro	id Virtual Device							
	AVD Name	Target Name		Platform	API Level	CPU/ABI		Details	
	droid	Android 2.3.3	1	2.3.3	10	ARM (armeabi	)	Start	
								Refresh	

You should now see something like this:



# Conclusion

Congratulations you have successfully created an Android application. While this application note taught you how to create a basic application you can use the techniques you learned to expand on your application and turn it into something more powerful.

### Resources

http://www.developer.android.com http://www.eclipse.org/downloads/ http://www.oracle.com/technetwork/java/javase/downloads/index.html