Android sdk

Stands for "Software Development Kit." An SDK is a collection of software used for developing applications for a specific device or operating system.

Android Virtual Device (AVD)

An *Android Virtual Device* (AVD) is a device configuration that is run with the Android emulator. It works with the emulator to provide a virtual device-specific environment in which to install and run Android apps. Lesson 4 shows you how to create an AVD by introducing you to the Android SDK's AVD Manager tool.

Android - Emulator

The Android SDK includes a virtual mobile device emulator that runs on your computer. The emulator lets you prototype, develop and test Android applications without using a physical device.

In this chapter we are going to explore different functionalities in the emulator that are present in the real android device.

Creating AVD

If you want to emulate a real device, first crate an AVD with the same device configurations as real device, then launch this AVD from AVD manager.

Changing Orientation

Usually by default when you launch the emulator, its orientation is vertical, but you can change it orientation by pressing Ctrl+F11 key from keyboard.

First launch the emulator. It is shown in the picture below -



Once it is launched, press Ctrl+F11 key to change its orientation. It is shown below

Sr.No	Command & description
1	Home Shifts to main screen
2	F2 Toggles context sensitive menu
3	F3 Bring out call log
4	F4 End call
5	F5 Search
6	F6 Toggle trackball mode
7	F7 Power button
8	F8 Toggle data network
9	Ctrl+F5 Ring Volume up
10	Ctrl+F6 Ring Volume down



Emulator Commands.

Apart from just orientation commands, there are other very useful commands of emulator that you should keep in mind while using emulator. They are listed below –

Emulator - Sending SMS

You can emulate sending SMS to your emulator. There are two ways to do that. You can do that from DDMS which can be found in Android studio, or from Telnet.(Network utility found in windows).

Sending SMS through Telnet.

l.	Windows Features 🗧 🗖	×
Turn	Windows features on or off	•
fo turr heck l	n a feature on, select its check box. To turn a feature off, clear its box. A filled box means that only part of the feature is turned on	
ŧ 🗌	🤰 Microsoft Message Queue (MSMQ) Server	^
•	Print and Document Services	
~	Remote Differential Compression API Support	
	📕 RIP Listener	100
ŧ 🗌	Simple Network Management Protocol (SNMP)	
	Ji Simple TCPIP services (i.e. echo, daytime etc)	
~	SMB 1.0/CIFS File Sharing Support	
~	🕌 Telnet Client	
-	👔 Telnet Server	
	TETD Client	
	Windows Identity Foundation 3.5	
1	Windows Location Provider	Y
	OK Cancel	1
	OK	

Telnet is not enabled by default in windows. You have to enable it to use it. Once enabled you can go to command prompt and start telnet by typing telnet.

In order to send SMS, note down the AVD number which can be found on the title bar of the emulator. It could be like this 5554 e.t.c. Once noted, type this command in command prompt.

telnet localhost 5554

Press enter when you type the command. It is shown below in the figure.



You will see that you are now connected to your emulator. Now type this command to send message.

sms send 1234 "hello"

Once you type this command, hit enter. Now look at the AVD. You will receive a notification displaying that you got a new text message. It is shown below –



Emulator - Making Call

You can easily make phone calls to your emulator using telent client. You need to connect to your emulator from telnet. It is discussed in the sending sms topic above.

After that you will type this command in the telent window to make a call. Its syntax is given below –

gsm call 1234

Once you type this command , hit enter. Now look at the AVD. You will receive a call from the number your put in the command. It is shown below –



Emulator - Transferring files

You can easily transfer files into the emulator and vice versa. In order to do that, you need to select the DDMS utility in Android studio. After that select the file explorer tab. It is shown below –

0			Andr	old Device	Monit	ore			
File Edit Run Window Help									
						Quick Access		E ODMS	11-5-0 日日 4
Devices 10	- 0	🗟 Threads 🗑 He	tap 🔒	Allocation T	The No	etwork Sta	File Explore	e Si 🕥 Emulato	e C 🔲 System info 📟 🖸
* 6 6 6 5 5 0	3 C # V								10 +
al i sue con concernen. Al	and a second second	Name	Size	Date	Time	Permissions	info		
Mana		N De arct		2015-04-01	09.47	dryner-se-x			
	11-12-1	> (a cache		2015-04-01	13:55	dtworwx			
. In News 2, APL, Online	Nerus 5	h charger		1970-01-01	05:30	Invorteories.	-> /sbin/h		
comlandri 1472	8500	p 🕞 config		2015-04-01	09:47	dr-x			
com.goog 3011	8601	(a) d		2015-04-01	09:47	Inwanwarawa.	-> /sys/ket-		
com.goog 1700	8602	> 25 data		2015-03-30	14.40	dramarwa-x			
comandel 1318	8003	default.pro	281	1970-01-01	05:30	-IM-I+-I++			
comandri 1445	0004	- b 😂 dev		2015-04-01	09:47	dram-ar-x			
comanan 1435	0003	@ etc		2015-04-01	09:47	Invaniana	-> /system_		
system_pr_12.50	80.0	file_contex	11013	1970-01-01	05:30	-rw-t-+r			
comanon 3009	0007	fstab.goldf	922	1970-01-01	05:30	- [W-1			
android pr 2547	0000	init init	500228	1970-01-01	05:30	-PACET-IE			
comutation 1780	00/9	init.enviror	981	1970-01-01	05:30	-1403-8			
comandri 2996	8610	init.goldfis	2836	1970-01-01	05:30	-rwst-x			
com.goog 1/20	8911	init.rc	21982	1970-01-01	05:30	-FADEL-K			
comandri 2057	8612	init.trace.rr	1927	1970-01-01	05:30	-rwht-x			
com.goog 1890	8015	init.usb.rc	3885	1970-01-01	05:30	-rwst-x			
comunder 1575	3014	int.zygote	301	1970-01-01	05.30	-fw0d-a			
comigoog 1759	8015	> (= mot		2015-04-01	09-47	drynmint-x			
android.pr. 2783	3075	> On proc		2015-04-01	09.47	drarara.			
		B property c	2614	1970-01-01	05:30	-(W-D-1			
		b Ca root		2014-09-16	23-03	drets			
		- P Co shin		1970-01-01	05.30	diwar-a			
		C= sdcard		2015-04-01	09.47	Invenience	-> /storag_		

Browse through the explorer and make new folder, view existing contents e.t.c.

Dalvik Virtual Machine | DVM

As we know the modern JVM is high performance and provides excellent memory management. But it needs to be optimized for low-powered handheld devices as well.

The **Dalvik Virtual Machine (DVM)** is an android virtual machine optimized for mobile devices. It optimizes the virtual machine for *memory*, *battery life* and *performance*.

Dalvik is a name of a town in Iceland. The Dalvik VM was written by Dan Bornstein.

The Dex compiler converts the class files into the .dex file that run on the Dalvik VM. Multiple class files are converted into one dex file.

Let's see the compiling and packaging process from the source file:



The **javac tool** compiles the java source file into the class file.

The **dx tool** takes all the class files of your application and generates a single .dex file. It is a platform-specific tool.

The **Android Assets Packaging Tool (aapt)** handles the packaging process.

Runtime Environment

Runtime Environment consists of software instructions(generated from the code) that are used while executing the programming.

JVM is the component that is used to convert bytecode into machine code in order to run Java-based programs.

Why is a Virtual Machine need to run any Application?

- A Virtual Machine isolates the execution of the program from the OS. Thus protecting malicious code from affecting the system files.
- Virtual Machines execute code independent of the CPU architecture

Dalvik Virtual Machine (DVM) was specifically designed to run Android applications initially.

Why Android uses DVM and not JVM?

Mobile Environment is not as powerful as your computer systems(majorly). There are battery and ram constraints. DVM was specifically optimized in order to run on Android.

Mobile Environment is not as powerful as your computer systems(majorly). There are battery and ram constraints. DVM was specifically optimized in order to run on Android.

Android DVM

Before looking at the Dalvik Virtual Machine, let's look at the JVM:

Following is the flow of JVM:



Following is the flow of DVM:



We know that DVM is specifically used for low memory devices.

The Dex compiler (dx tool) converts the .class files generated from the javac compiler to .dex file.

These .dex files are then converted to machine code.

Note: **dexopt** tool which is a part of the DVM converts the dex file into .odex file format.

JVM runs the bytecode of pure java classes while DVM runs the bytecode of the dex formatted .dex file that was recompiled from the Java bytecode.

JVM dynamically loads the bytecode for each class from the corresponding .class file. While Dalvik bytecode is only composed of one .dex file, containing all the classes of the application.

How is the dex bytecode converted into machine code?

Using **JIT**(Just In Time).

Just In Time is a component that takes application code, analyzes it, and actively translates it into a form that runs faster, doing so while the application continues to run. This leads to increased launch time for applications since it needs to be done everytime the application is launched.

As the execution progresses, more bytecode is compiled and cached. This leads to faster boot times.

DVM and JIT were replaced by ART and AOT respectively since Android Lollipop.



Step 1 - System Requirements

You will be delighted, to know that you can start your Android application development on either of the following operating systems –

- Microsoft® Windows® 10/8/7/Vista/2003 (32 or 64-bit)
- Mac® OS X® 10.8.5 or higher, up to 10.9 (Mavericks)
- GNOME or KDE desktop

Second point is that all the required tools to develop Android applications are open source and can be downloaded from the Web. Following is the list of software's you will need before you start your Android application programming.

• Java JDK5 or later version

- Java Runtime Environment (JRE) 6
- Android Studio

Step 2 - Setup Android Studio

Overview

Android Studio is the official IDE for android application development. It works based on **IntelliJ IDEA**, You can download the latest version of android studio from <u>Android</u> <u>Studio 2.2 Download</u>, If you are new to installing Android Studio on windows, you will find a file, which is named as *android-studio-bundle-143.3101438-windows.exe*. So just download and run on windows machine according to android studio wizard guideline.

If you are installing Android Studio on Mac or Linux, You can download the latest version from <u>Android Studio Mac Download</u>, or <u>Android Studio Linux Download</u>, check the instructions provided along with the downloaded file for Mac OS and Linux. This tutorial will consider that you are going to setup your environment on Windows machine having Windows 8.1 operating system.

Installation

So let's launch *Android Studio.exe*,Make sure before launch Android Studio, Our Machine should required installed Java JDK. To install Java JDK,take a references of <u>Android environment setup</u>



Once you launched Android Studio, its time to mention JDK7 path or later version in android studio installer.

	Android Studio Setup	- 🗆 🗙				
2	Verifying your system meets the minim System Check	um requirements				
We could not detect a browse to its path if k	We could not detect a Java Development Kit (JDK) v7 or newer on your system. Please browse to its path if known:					
		Browse				
or download the foll and press 'Next' aft	lowing compatible JDK: jdk-7u67-windows-x64.ex	2				
	< Back Next	> Cancel				

Below the image initiating JDK to android SDK

	Android Stud	lio Setup		
	Browse For	Folder	×	irements
Brou	vse to a JDK location			
Ne could no				lease
	🏾 🎒 Java			owse
or downloa	a jdk1.7.0_75			
	bin bin			
	Include			
	Þ 퉲 jre			
	▷ 🏭 lib ▷ 🛄 ire7		~	
M	ake New Folder	ок	Cancel	
	Ľ.	< Back	Next >	Cancel

Need to check the components, which are required to create applications, below the image has selected Android Studio, Android SDK, Android Virtual Machine and performance(Intel chip).

	Android Studio Setup 🚽 🗖 🗙				
	Choose Components Choose which features of Android Studio you want to install.				
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.					
Select components to install:	 ✓ Android Studio ✓ Android SDK ✓ Android Virtual Device ✓ Performance (Intel® HAX 	Description Position your mouse over a component to see its description.			
Space required: 3.8GB	< >				
< Back Next > Cancel					

Need to specify the location of local machine path for Android studio and Android SDK, below the image has taken default location of windows 8.1 x64 bit architecture.

	Android Studio Setup		×
2	Configuration Settings Install Locations		
Android Studio The location Click Browse	Installation Location specified must have at least 500MB of free space. to customize:		
C:\Program	Files\Android\Android Studio	Browse	
The location Click Browse	specified must have at least 3.2GB of free space. to customize:		
C:\Users\sa	aira_000\AppData\Local\Android\sdk	Browse	
	< Back Next >	Cance	ł

Need to specify the ram space for Android emulator by default it would take 512MB of local machine RAM.

	Android Studio Setup 🚽 🗖 🗙					
2	Configuration Settings Emulator Setup					
We have detected the performance mode.	We have detected that your system can run the Android emulator in an accelerated performance mode.					
Please set the maxin Manager (HAXM) to	num amount of RAM available for the Intel Hardware Accelerated use for all x86 emulator instances.					
You can change thes for more information	You can change these settings at any time. Please refer to the Intel HAXM Documentation for more information.					
Recommended:	512 MB					
O Custom:	512 MB V					
	* This value must be between 512 MB and 1 GB					
Note: Setting aside a large memory reservation may cause other programs to run slowly when using the x86 Android emulator with HAXM.						
	< Back Next > Cancel					

At final stage, it would extract SDK packages into our local machine, it would take a while time to finish the task and would take 2626MB of Hard disk space.

	Android Studio Setup	□ ×
R	Installing Please wait while Android Studio is being installed.	
Extracting Android SDK	4% (108 / 2626 MB)	
Extract: terminal.jar. Output folder: C:\Pro Output folder: C:\Pro Extract: resources_e Extract: testng-plugin Extract: testng.jar Output folder: C:\Pro Output folder: C:\Use Extract: android-sdk. Output folder: C:\Use	100% gram Files \Android \Android Studio \plugins \testng gram Files \Android \Android Studio \plugins \testng \ib n.jar 100% 100% gram Files \Android \Android Studio ers \SAIRA_~1 \AppData \Local \Temp 7z ers \saira_000 \AppData \Local \Android \sdk	^
	< Back Next >	Cancel

After done all above steps perfectly, you must get finish button and it gonna be open android studio project with Welcome to android studio message as shown below



You can start your application development by calling start a new android studio project. in a new installation frame should ask Application name, package information and location of the project.

R	Create New Project	×
New Android S	Project	
Configure you	r new project	
Application name:	0	_
Company Domain:	saira_000.example.com	-1
Package name:	com.example.saira_000.	Edit
Project location:	C/\Users\saira_000\AndroidStudioProjects	-
Please enter an applic	cation name (shown in launcher)	
	Previous Next Cancel Fin	ish

After entered application name, it going to be called select the form factors your application runs on, here need to specify Minimum SDK, in our tutorial, I have declared as API23: Android 6.0(Mashmallow)

🕐 Create New Project			×
Reget Android De	vices		
Select the form factors your app w	rill run on		
Different platforms may require separate SDKs			
🗾 Phone and Table	e		
Minimum SDK	API 23: Android 6.0 (Marshmallow)		
	Lower API levels target more devices, but have fewer features available.		
	By targeting API 23 and later, your app will run on approximately 4.7% of the devices that are active on the Google Play Store.		
	Help me choose		
🗌 Wear			
Minimum SDK	API 21: Android 5.0 (Lollipop)		-
Π τν			
Minimum SDK	API 21: Android 5.0 (Lollipop)		
C Android Auto			
Glass			
Minimum SDK	Glass Development Kit Preview (API 19)		•
		Previous Next	Cancel
		Transmission in the second sec	termination of the second seco

The next level of installation should contain selecting the activity to mobile, it specifies the default layout for Applications



At the final stage it going to be open development tool to write the application code.

Average (CLARE) CLARE) CLARE (CLARE) CLARE (CLARE) CLARE) CLARE (CLARE) CLARE (CLARE) CLARE) CLARE (CLARE) CLARE (CLARE) CLARE) CLARE) CLARE (CLARE) CLAR	a (2) mente () (2) mente anti () 20 ft () (2) Madridea para () (2) general ()	
Cale Constant Cale Constant Cale Constant	Notes 0 - 1 Image: Control of the contro of the control of the control of the control of the contr	AL: Conservent here I ♦ 9- 4 Conservent here I ♦ 50000 (over # Conservent here All Tophere - promptielle, well
	Kores Kardi Astron Kado Astron	Properties 2 D T Report Fields strive accessibility, califorgiant data Back prevent Back preve

Step 3 - Create Android Virtual Device

To test your Android applications, you will need a virtual Android device. So before we start writing our code, let us create an Android virtual device. Launch Android AVD Manager Clicking AVD_Manager icon as shown below

800	activity, my and - Tapel - Cher Rodrich, 1-/Programming/OMCAndroid - Android	id Shid	to sterio	0.8.2				1
DHØ XGB Q Q +- 1	**** * * L * A * A * 7							9
C. OliCAndroid (C. and (C. and (C. main, C. en)))	ayout) arthing, my arel And Marager							
E.Propez	Welchelaus = Santy my att m		Pressor					01.11
 CollCAndread 1- Programming/CollCAndread) Endes En	<pre>*#elatizetaput milisizetaput **tip://sibess.android.com/aptrox/android* androita/signal_delata_minit_parts: android/space_delata_minit_parts: android/space_delata_minit_parts: android/space_delata_minit_parts: android_parts: android_parts: android_parts: android_space_delata_minit_parts: android_part</pre>		Q 10-	Result 4-	AppThene AppThene Contraction		0. #L < 0	
P hold made	Design Text							
heming 2013-00 Gradie build finished in 3 sec 7 7								9-1
Caralle built fresheel in 3 one 13 metates agai						171 151	10164 114	-

After Click on a virtual device icon, it going to be shown by default virtual devices which are present on your SDK, or else need to create a virtual device by clicking **Create new Virtual device** button

👧 Andro	id Virtual Device Manager						-	0	3	\times
2	Your Virtual Devi Andreid Studio	ces								
VT-x is	disabled in BIOS.							Trout	lesho	at
Type	Name	Resolution	API	Target	CPU/ABI	Size on Disk	Ac	tions		
	Nexus SX API 23	1080 × 1920: 420dpi	23	Android 6.0 (Google APh)	x86	2 GB		•	/ -	-
+ Cr	eate Virtual Device									ø

If your AVD is created successfully it means your environment is ready for Android application development. If you like, you can close this window using top-right cross button. Better you re-start your machine and once you are done with this last step, you are ready to proceed for your first Android example but before that we will see few more important concepts related to Android Application Development.

Hello Word Example

Before Writing a Hello word code, you must know about XML tags. To write hello word code, you should redirect to **App>res>layout>Activity_main.xml**

▼	🕞 арр
	manifests
	🕨 🛅 java
	V 📴 res
	💼 drawable
	Iayout
	activity_main.xml
	menu
	🕨 💼 mipmap
	values
►	Gradle Scripts

To show hello word, we need to call text view with layout (about text view and layout, you must take references at <u>Relative Layout</u> and <u>Text View</u>).



Need to run the program by clicking **Run>Run App** or else need to call **shift+f10**key. Finally, result should be placed at Virtual devices as shown below

