Android development guide

Revision	Date	Description
V1.0	2016/7/15	Android development guide

Archermind

2016/7/19

Contents

1. Introduction

If you already have the android development environment or the ability of building the environment, you could skip the chapter 2.1, and continue to see the rest of the content.

2. Prepare

2.1. Environment building

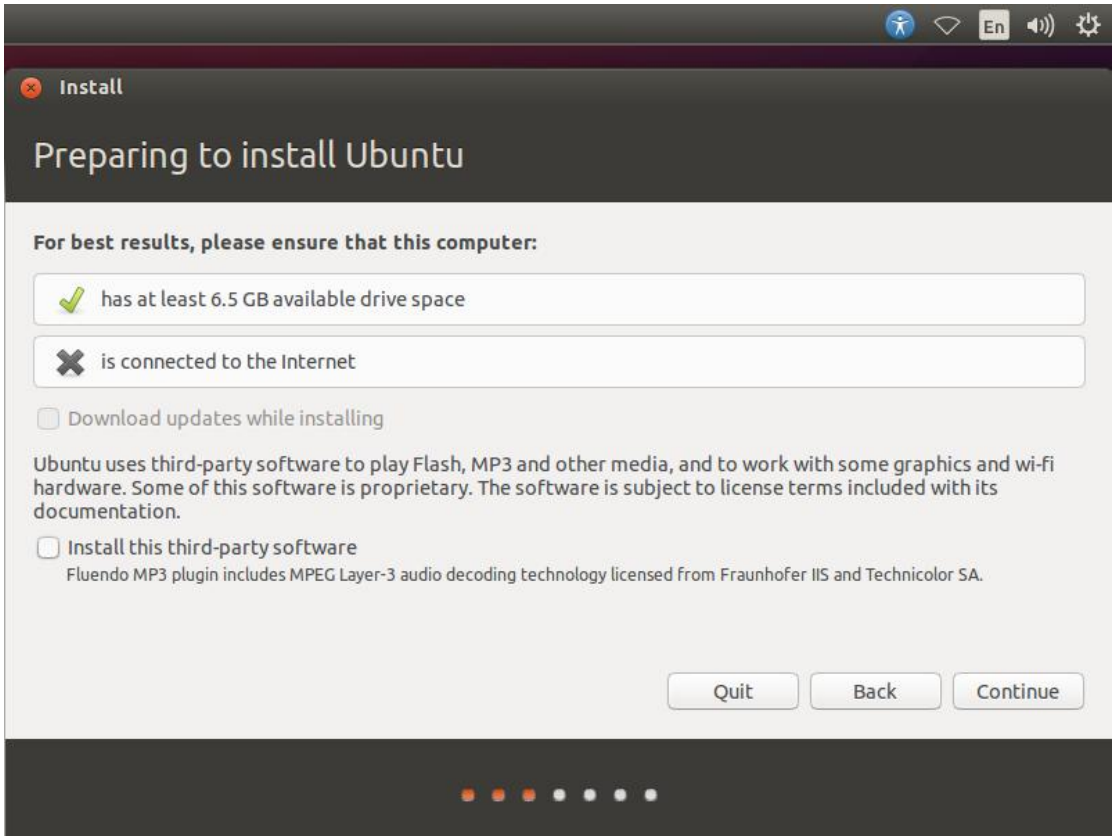
2.1.1. Ubuntu installation

Recommend OS: Ubuntu 14.04(LTS) 64bit

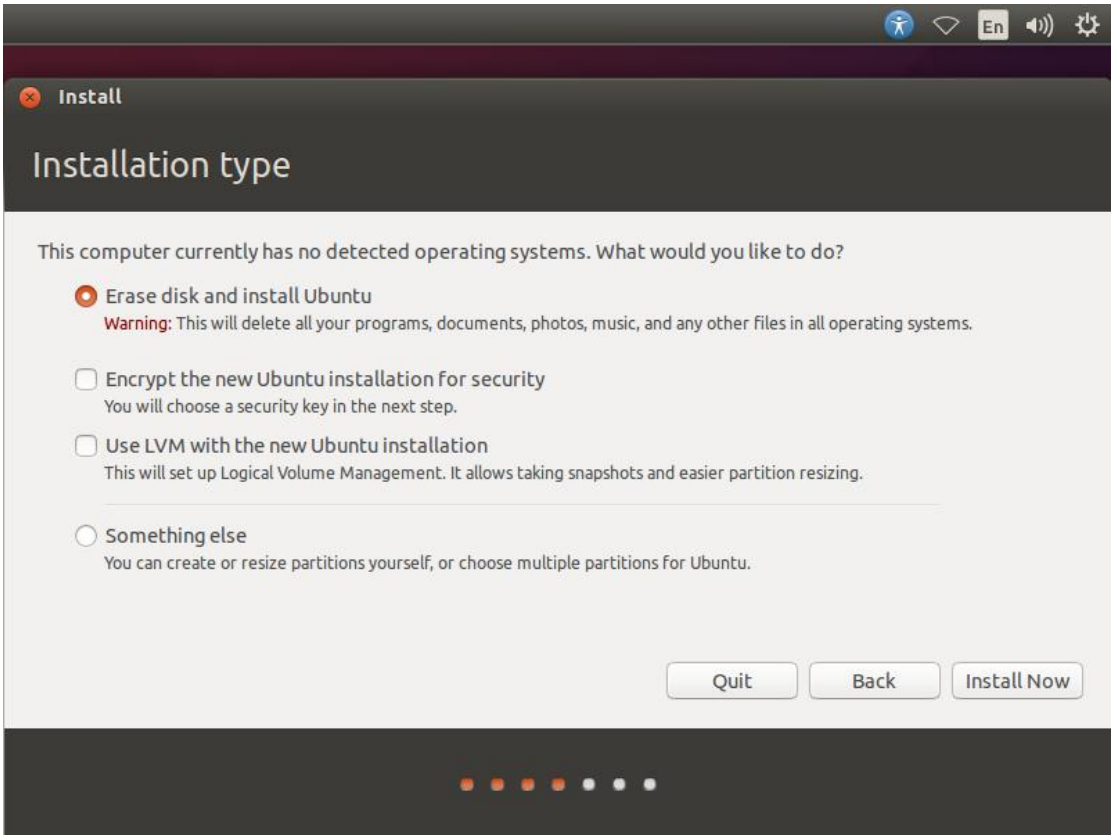
Download address: http://releases.ubuntu.com/14.04/

Installation steps:

1) Start to install:

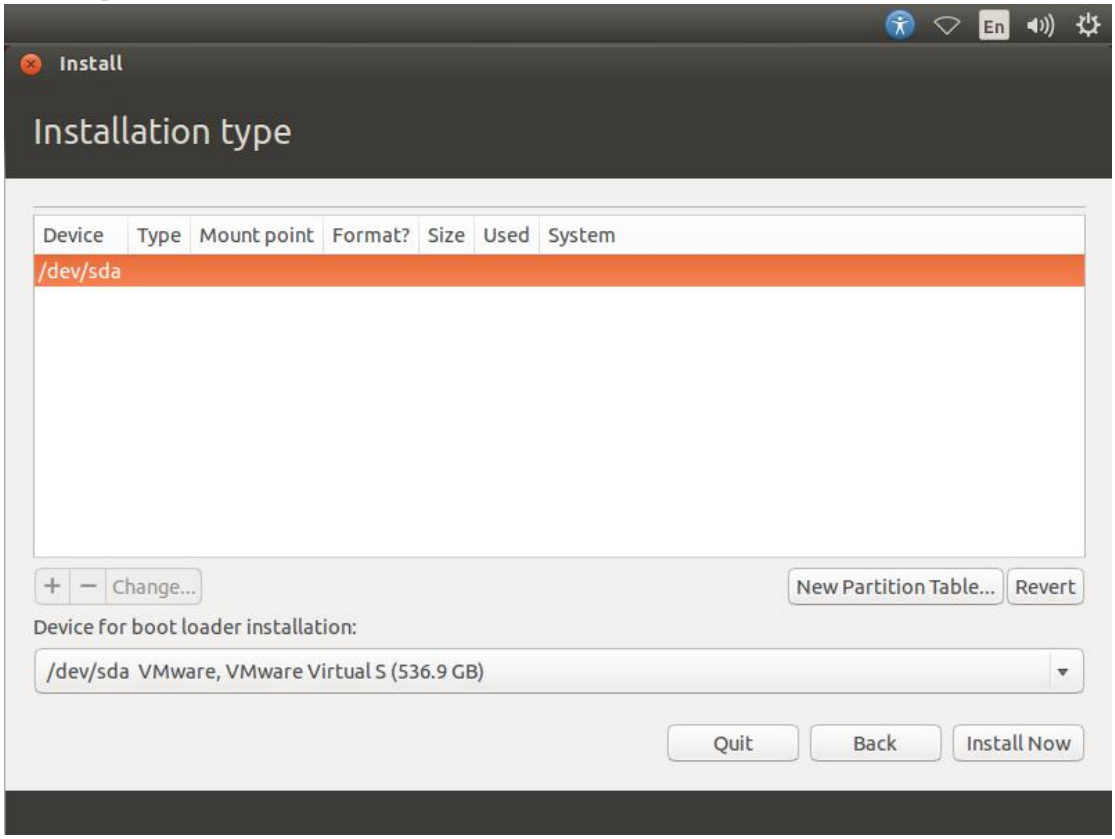


2) Choose installation type:



You should choose your installation type base-on your server or machine. You could choose "Something else" if you want to customize the partitions.

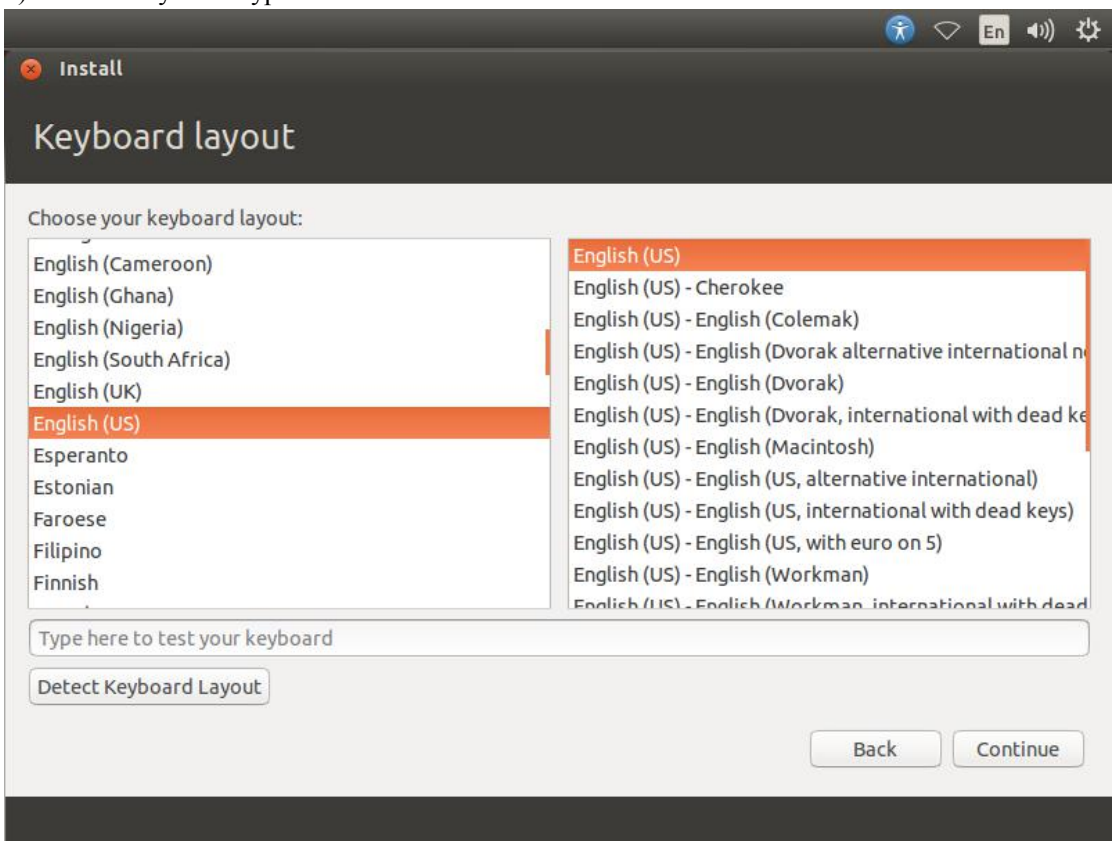
3) Disk partition:



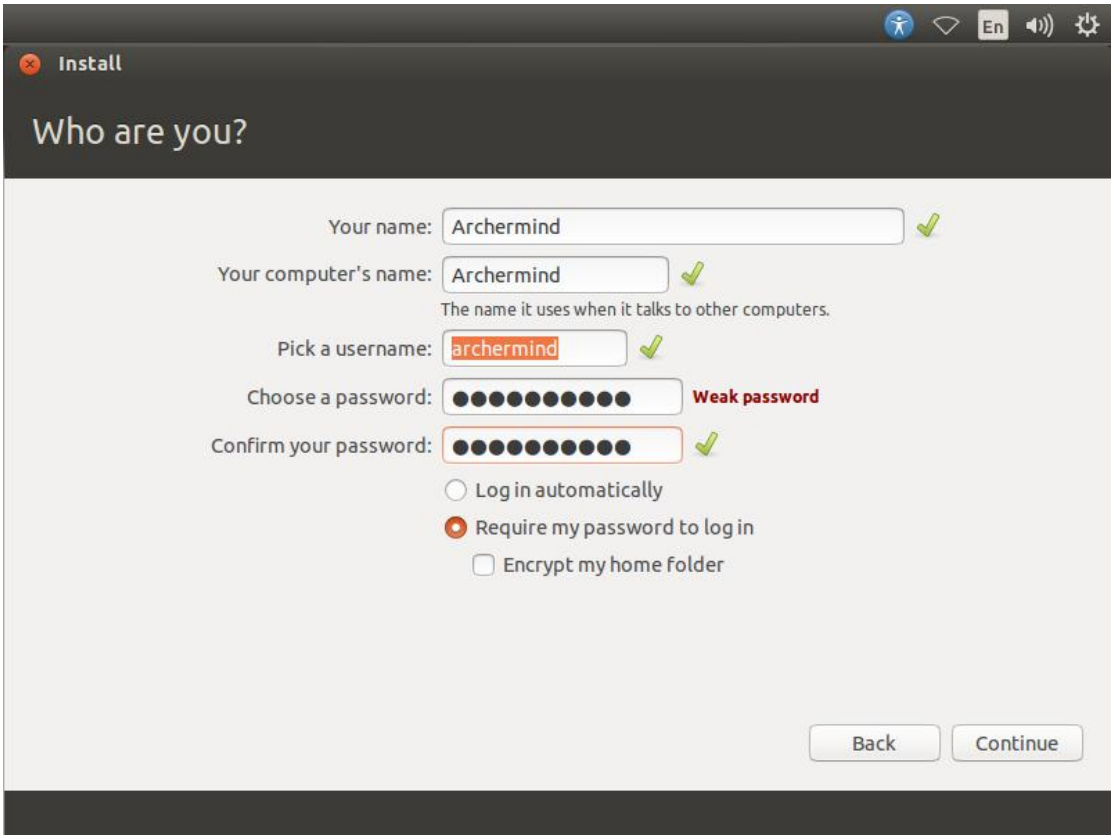
If you choose "Something else" on above step, you should customize the partitions by yourself.

4) Choose timezone:

5) Choose keyboard type:

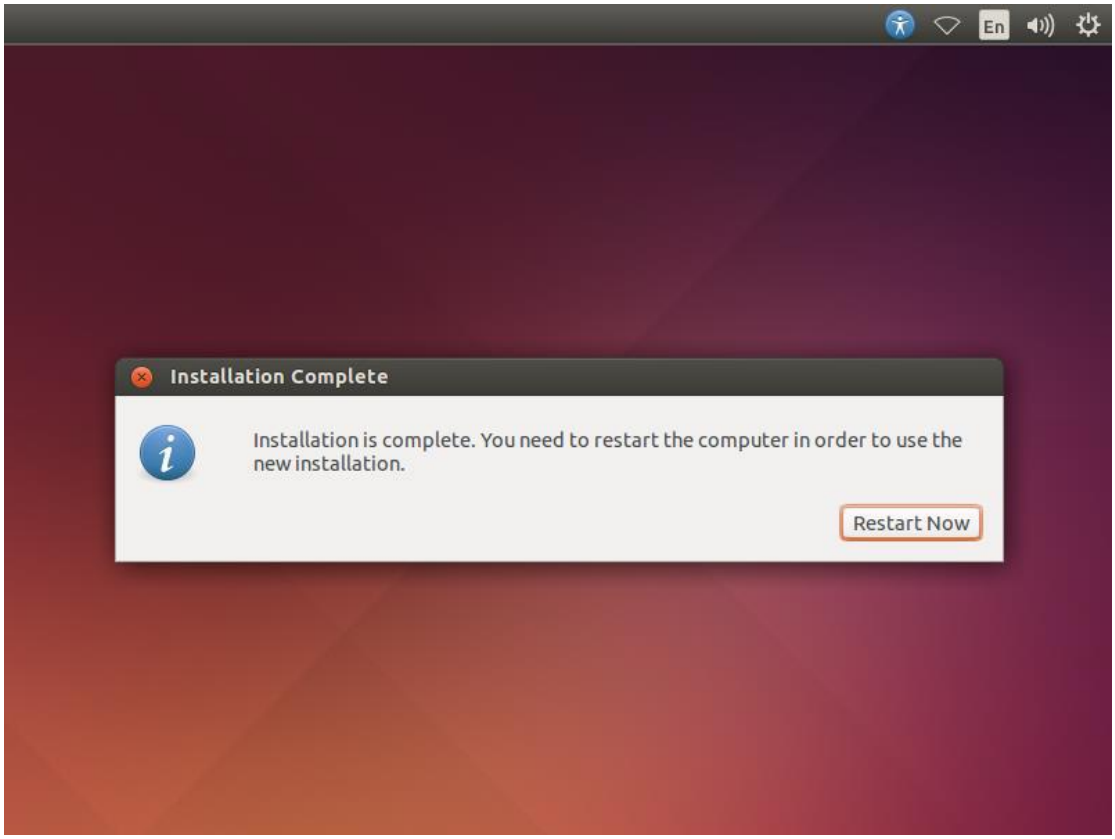


6) Fill in information:



7) Installing...:

8) Installation complete:



When the installation is complete, please reboot your machine.

2.1.2. Ubuntu tools installation

Open a terminal, and execute the following command:

```
$ sudo apt-get install git-core gnupg flex bison gperf libsdl1.2-dev \
 libesd0-dev libwxgtk2.8-dev squashfs-tools build-essential zip curl \
 libncurses5-dev zlib1g-dev pngcrush schedtool libxml2 libxml2-utils \
 xsltproc lzop libc6-dev schedtool g++-multilib lib32z1-dev lib32ncurses5-dev \
 lib32readline-gplv2-dev gcc-multilib libswitch-perl
```

2.1.3. JDK installation(Linux)

1）Installation:

```
$ sudo apt-get install default-jdk
$ sudo apt-get install default-jre
```

2）Configuration JDK Environment:

Edit .bashrc file, and add the following settings to the end of the file:

```
export JAVA_HOME=/usr/java/jdk1.6.0_10    （should be same with the JDK folder name）
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export PATH=$PATH:$JAVA_HOME/bin
```

If you want to make the change take effect immediately, please execute the following command in the terminal:

```
$ source ~/.bashrc
```

3） test:

```
$ java -version
```

2.1.4. Repo installation

1) Ensure there is a ~/bin directory in your home director:

```
$ mkdir ~/bin
```

2) Download repo script:

```
$ curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
```

3) Add executable permissions to repo

```
$ chmod a+x ~/bin/repo
```

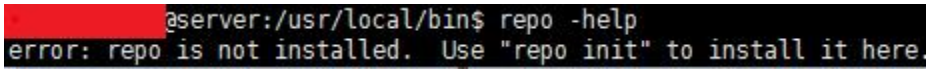
4) Include the installation directory of repo in your path:

```
$ export PATH=~/bin:$PATH
```

5) Run repo --help to verify the installatio:

```
$ repo -help
```

You could see a message similar to the following:



Common commands:

```
init -> install repo in current directory

help -> display the details on a command
```

You could also install the repo like the url: https://source.android.com/source/downloading.html.

2.1.5. JDK installation(Windows)

1) Installation steps:

Download the appropriate version from the website:

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

Then double click the .exe file to start the installation, and the installation wizard will guide you to finish the installation.

2） Configuration JDK Environment:

- Right click the My Computer icon, then click Properties -> Advanced -> Environment Variables;

- System variable -> New JAVA_HOME variable (Variable values: JDK installation directory. For example: c:\java\jdk1.7.0);
- System variable -> Path variable -> edit, append to start with: %JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;
- System variable -> New CLASSPATH variable (Variable values : .;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;)

3）test:

Use command **java -version** to verify the configuration, for example:

```
C:\Users\n006370>java -version
java version "1.8.0_66"
Java(TM) SE Runtime Environment (build 1.8.0_66-b18)
Java HotSpot(TM) 64-Bit Server VM (build 25.66-b18, mixed mode)
```

2.1.6. Eclipse installation and configuration

1) Eclipse download/installation

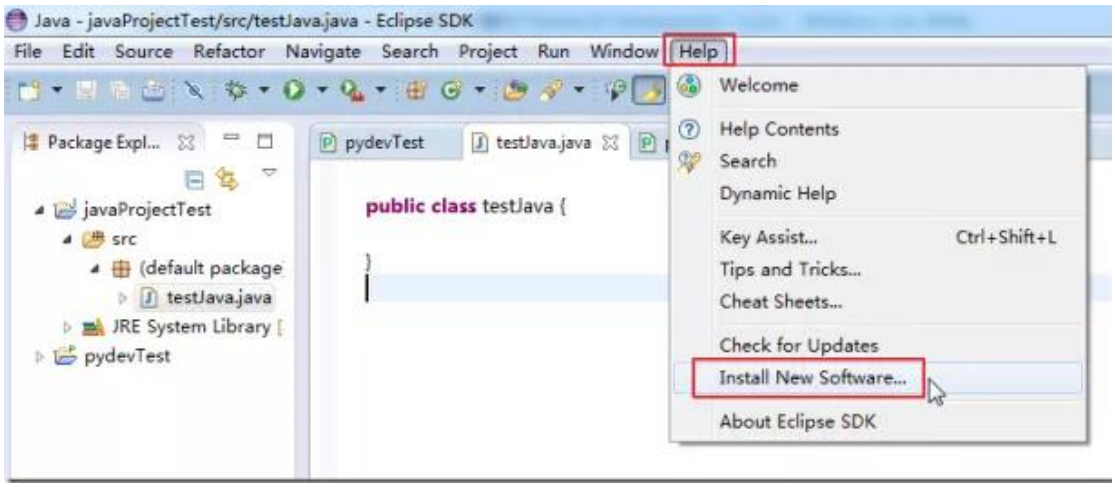
- Download the version from the website: https://www.eclipse.org/.
  Note: The downloaded version needs to be consistent with your operating system.
- Choose a directory to unzip the .zip file, then double click eclipse.exe to use the Eclipse.

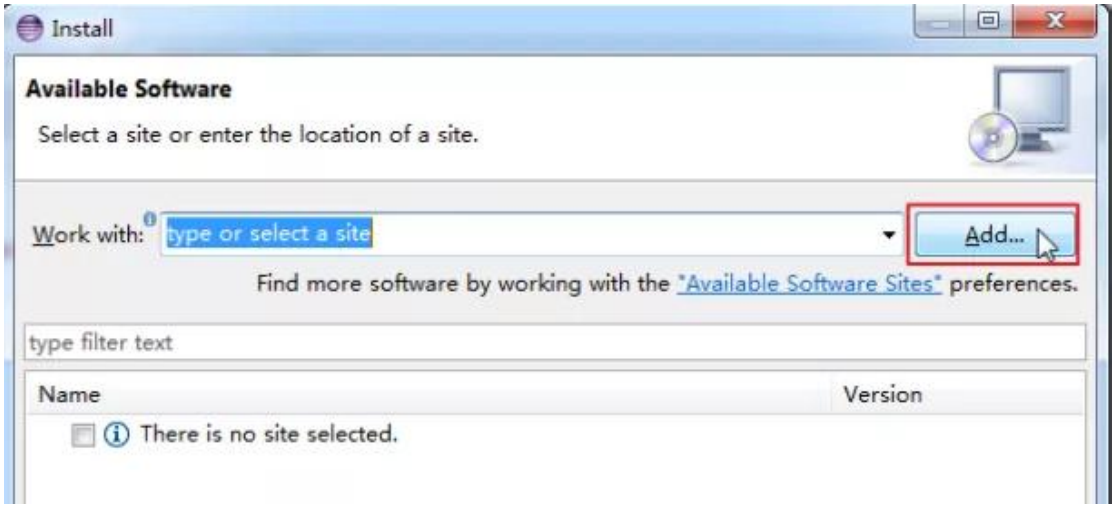
2）Eclipse environment configuration

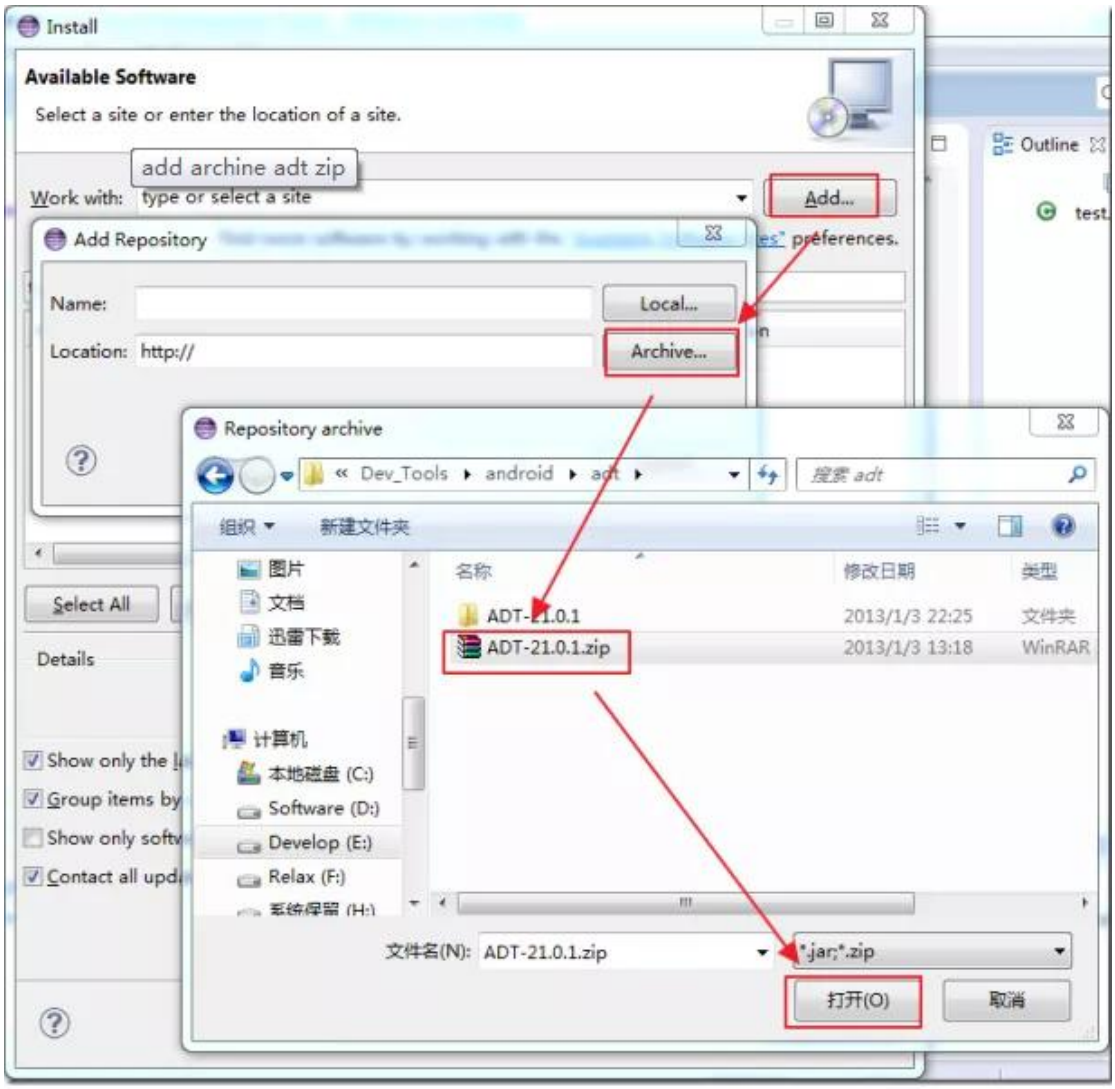
The installation of ADT has two ways: online installation and offline installation.

offline installation:

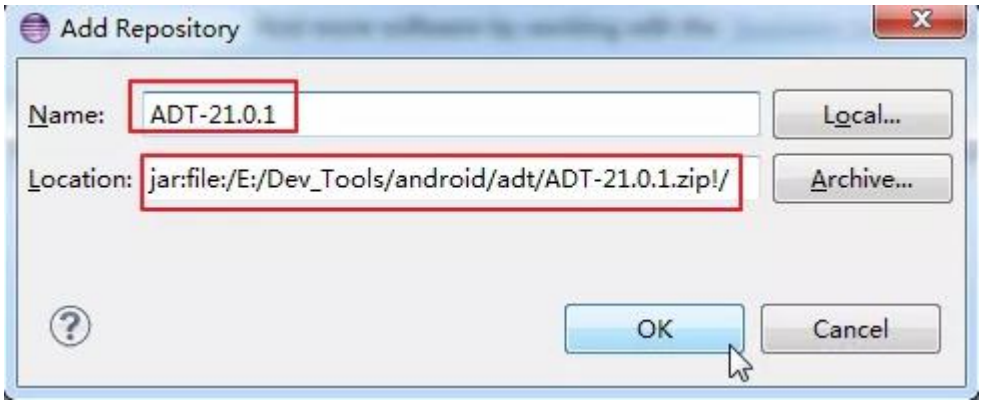
Open Eclipse, click Help -> Install New Software -> Add Repository -> Archive, choose the prepared ADT archive.

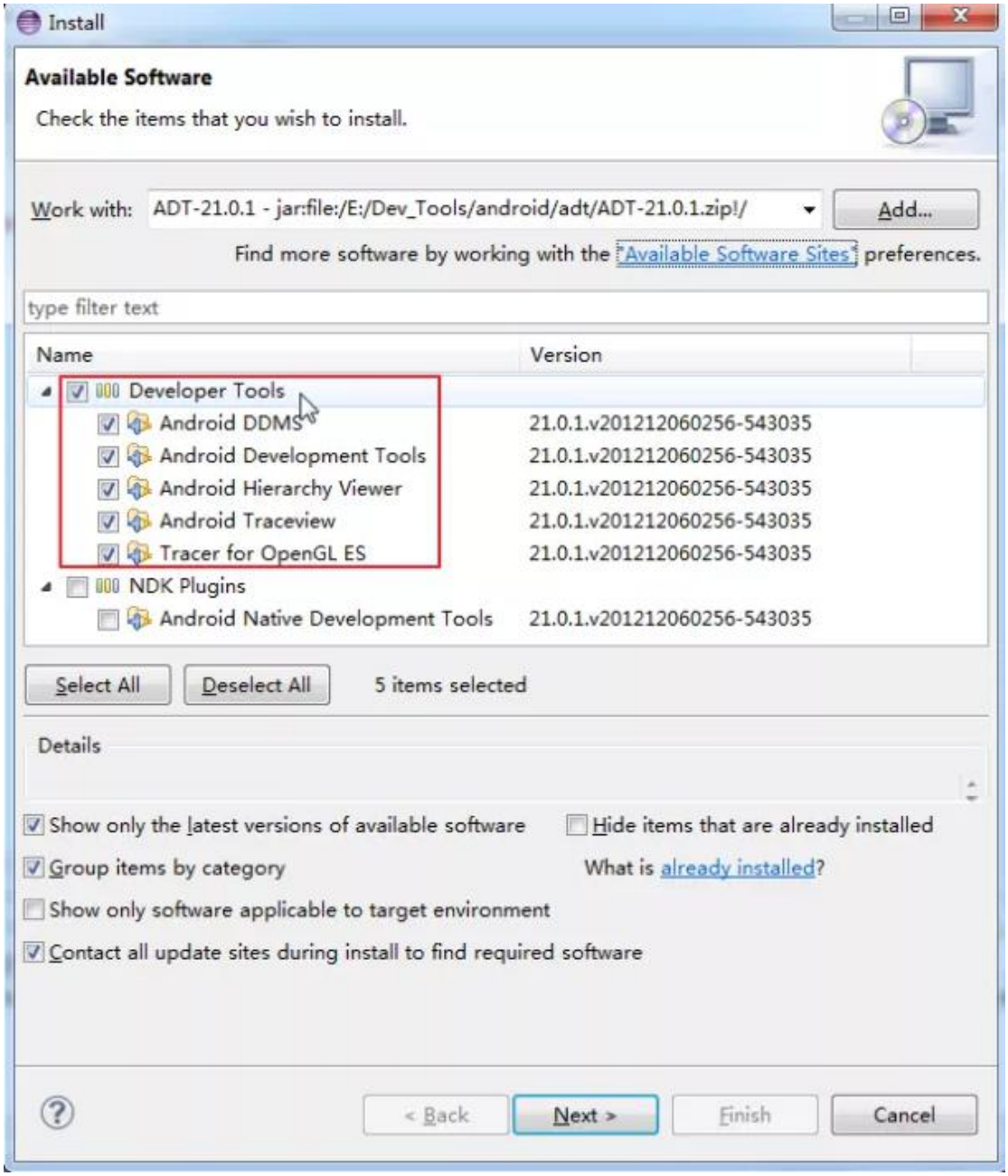


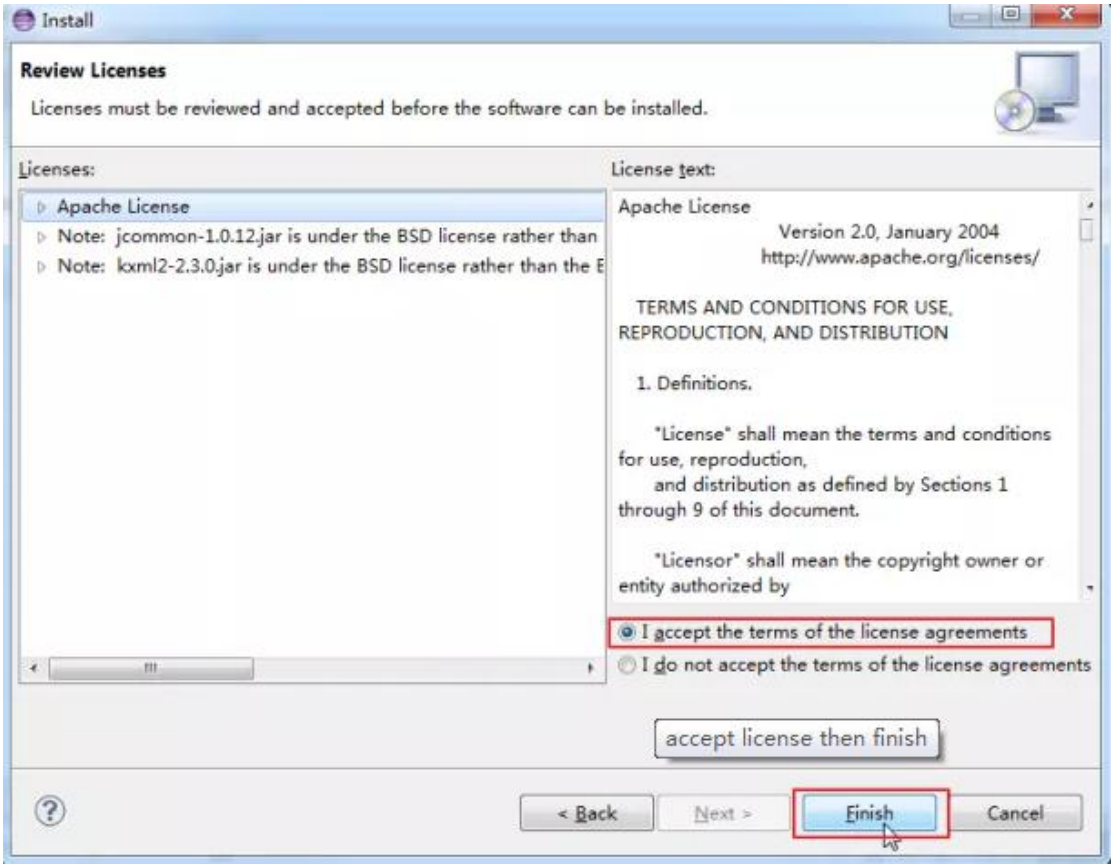




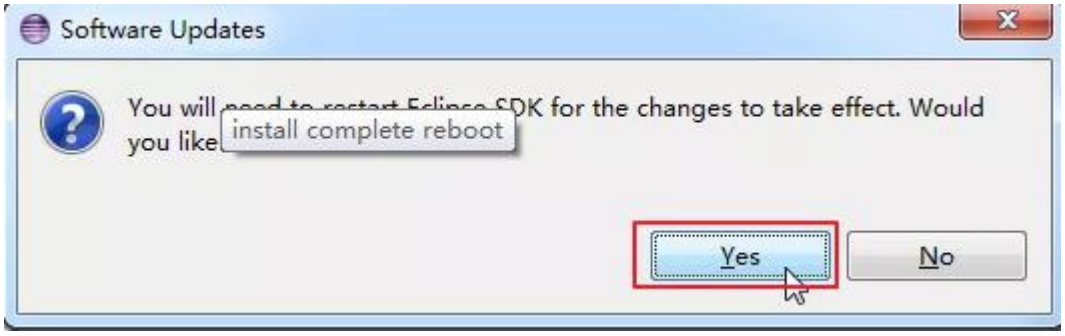
Add the name then click OK to return back to the install dialog, check the Developer Tools, then continue the installation until complete, then restart the Eclipse.


Add Repository
Name:
ADT-21.0.1
Local...
Location:
jar:file:/E:/Dev_Tools/android/adt/ADT-21.0.1.zip!/
Archive...
OK
Cancel


Install
Available Software
Check the items that you wish to install.
Work with:
ADT-21.0.1 - jar:file:/E:/Dev_Tools/android/adt/ADT-21.0.1.zip!/
Add...
Find more software by working with the "Available Software Sites" preferences.
type filter text
Name
Version
Developer Tools
Android DDMS
21.0.1.v201212060256-543035
Android Development Tools
21.0.1.v201212060256-543035
Android Hierarchy Viewer
21.0.1.v201212060256-543035
Android Traceview
21.0.1.v201212060256-543035
Tracer for OpenGL ES
21.0.1.v201212060256-543035
NDK Plugins
Android Native Development Tools
21.0.1.v201212060256-543035
Select All
Deselect All
5 items selected
Details
Show only the latest versions of available software
Hide items that are already installed
Group items by category
What is already installed?
Show only software applicable to target environment
Contact all update sites during install to find required software
< Back
Next >
Finish
Cancel


Install
Review Licenses
Licenses must be reviewed and accepted before the software can be installed.
Licenses:
License text:
Apache License
Note: jcommon-1.0.12.jar is under the BSD license rather than
Note: kxml2-2.3.0.jar is under the BSD license rather than the E
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.
"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by
I accept the terms of the license agreements
I do not accept the terms of the license agreements
accept license then finish
< Back
Next >
Finish
Cancel


Installing Software
Installing Software
Security Warning
Warning: You are installing software that contains unsigned content. The authenticity or validity of
securty warning ok
e established. Do you want to continue with the installation?
OK
Cancel
Details >>
Run in Background
Cancel
<< Details

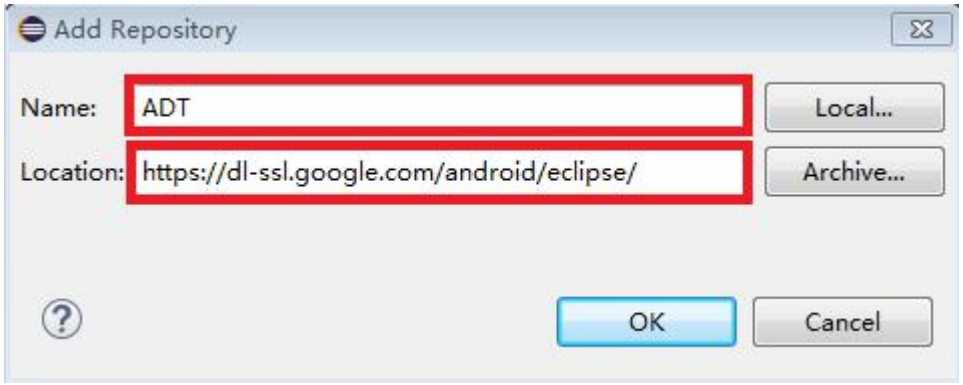


online installation:

The installation steps is similar with the offline installation, and the difference is: online installation needs to enter the URL address for ADT download, offline installation is to choose the ADT archive prepared already.

URL address:

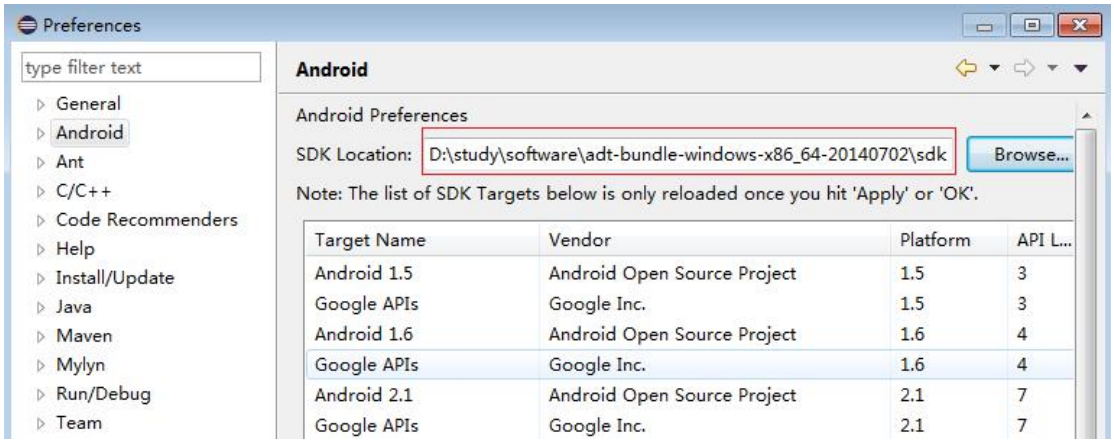
https://dl-ssl.google.com/android/eclipse/ or http://dl-ssl.google.com/android/eclipse/

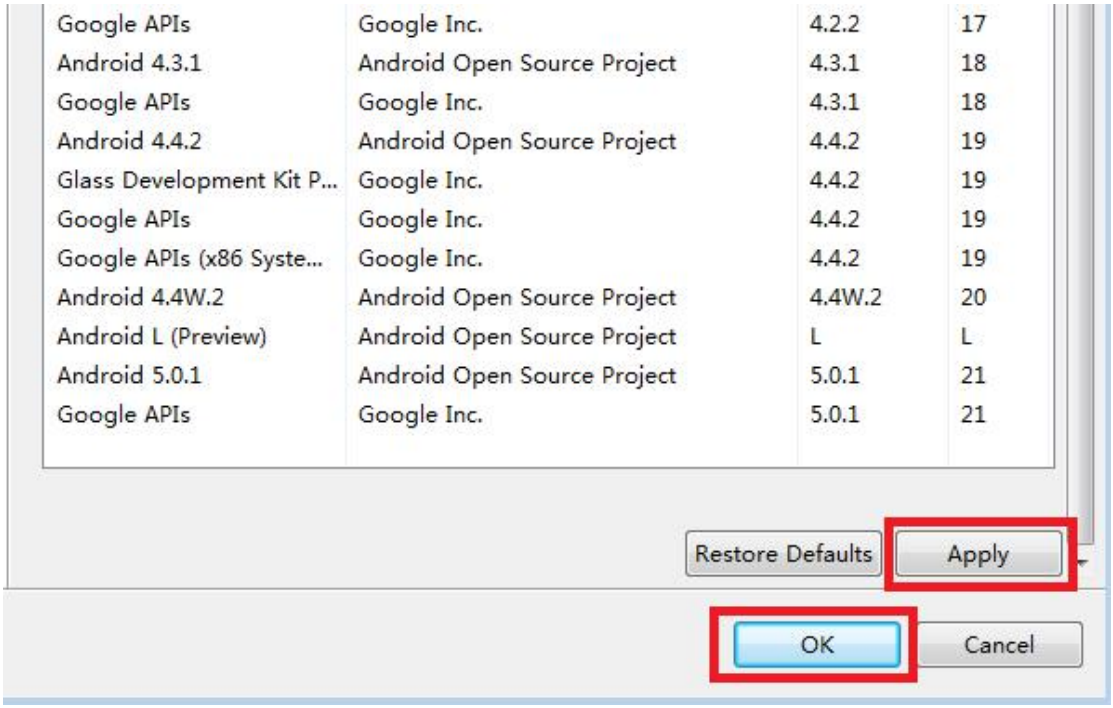


ADT configuration:

Extract the downloaded SDK Android to a directory.

Open Eclipse,choose Window -> Preferences -> Android -> SDK Location -> Browse,select the Android SDK directory, then click Apply and OK.





2.2. Code download path

We will update this part later.

3. Version compile and download

3.1. Android version compile

Step1. Enter into the root directory of the code:

```
$ source build/envsetup.sh
```

Step2. Version information:

full_XXXX-eng for eng version

full_XXXX-userdebug for debug version

full_XXXX-user for user version

```
$ lunch full_XXXX-eng
```

Step3. Build options:

3.1 Build all

```
$ make –j$(processes number)
```

3.2 Build preloader

```
$ make pl –j$(processes number)
```

3.3 Build lk

```
$ make lk –j$(processes number)
```

3.4 Build kernel

```
$ make kernel –j$(processes number)
```

3.5 Make boot image

```
$ make bootimage –j$(processes number)
```

3.6 Build android and make system image

```
$ make systemimage –j$(processes number)
```

3.2. Android version download

3.2.1. Necessary Condition

You need prepare 6 components(windows):

- xflash.exe(win) / xflash(linux)
- Normal load(Include image files and scatter file etc.)
- Special images and scatter file
- lib.cfg.xml
- fastboot.exe(win) / fastboot command(linux)

- fastboot command script file

3.2.2. Flash Tool access path

\vendor\mediatek\proprietary\system\core\xflash

3.2.3. How to build special images

Execute following commands, build system will automatically create FES folder and come out the special lk.bin, where FES store the needed files for xflash download to target befor entering fastboot mode.

```
$ source build/envsetup.sh
$ lunch full_amt6797_64_open-eng
$ make -j16 PLATFORM_FASTBOOT_EMPTY_STORAGE=yes -k 2>&1 | tee build.log
```

Then, you can find a folder named FES.
PATH: \out\target\product\amt6797_64_open\FES

3.2.4. Download

3.2.4.1. Windows Prepare

- xflash.exe
  \xflash\bin\win\xflash.exe
- Normal load(Include image files and scatter file etc.)
  You can put it in anywhere, eg, \xflash\bin\win\img
- Special images and scatter file
  You can put it in anywhere, eg, \xflash\bin\win\FES. How to build it?
  Please see “How to build special images”.
- lib.cfg.xml
  \xflash\bin\win\config
- fastboot.exe
  \xflash\bin\win, you shoud put it in normal load folder.
- fastboot command script file
  Writen by yourself, you should put it in normal load folder.

eg:fastboot command script file named xflash.bat

```
fastboot devices
fastboot flash recovery recovery.img
****
```

fastboot flash userdata userdata.img
fastboot reboot

3.2.4.2. Windows Download

Step 1. Make a device to enter fastboot mode

- Prepare special images and corresponding scatter file.
- Run program in command line mode like this:
  xflash.exe enter-fastboot “G:\xflash\bin\win\FES\MT6797_Android_scatter.txt”

```
G:\xflash\bin\win>xflash.exe enter-fastboot "G:\xflash\bin\win\FES\MT6797_Androi
d_scatter.txt"

START.
wait for device.
```

- Then plug in usb to device.
- Xflash will scan and open device COM port and connect it, download some necessary images to devices, then make device to enter fastboot mode.

```
connect boot rom.
boot from preloader.
download required partition images.
#1 write to lk
#2 write to logo
#3 write to tee1
jump to: tee1
END.
```

Step 2. Run fastboot command script file

- You need write a download script.
  Such as xflash.bat
  fastboot devices
  fastboot flash gpt PGPT
  fastboot flash preloader preloader_amt6797_64_open.bin
  fastboot flash recovery recovery.img
  fastboot flash scp1 tinysys-scp.bin
  fastboot flash scp2 tinysys-scp.bin
  fastboot flash lk lk.bin
  fastboot flash lk2 lk.bin
  fastboot flash boot boot.img
  fastboot flash logo logo.bin
  fastboot flash tee1 trustzone.bin
  fastboot flash tee2 trustzone.bin
  fastboot flash system system.img
  fastboot flash cache cache.img
  fastboot flash userdata userdata.img
  fastboot reboot
- Run the download script, download success.

3.2.4.3. Linux Prepare

- xflash
  \xflash\bin\linux\xflash
- Normal load(Include image files and scatter file etc.)
  You can put it in anywhere, eg, \xflash\bin\linux\img
- Special images and scatter file
  You can put it in anywhere, eg, \xflash\bin\linux\FES. How to build it?
  Please see “How to build special images”.
- lib.cfg.xml
  \xflash\bin\linux\config
- fastboot
  If your OS doesn't support fastboot command, pls install this cammand frist.
- fastboot command script file
  Writen by your self, you should put it in normal load folder.

3.2.4.4. Ubuntu Download

Step 1. Make a device to enter fastboot mode

- Prepare special images and corresponding scatter file.
- Run program in command line mode like this:
  sudo ./xflash enter-fastboot "/**/xflash/bin/win/FES/MT6797_Android_scatter.txt"
- Then plug in usb to device.
- Xflash will scan and open device COM port and connect it, download some necessary images to devices, then make device to enter fastboot mode.

Step 2. Run fastboot command script file

- You need write a download script.
  Such as xflash.sh

```
#!/bin/bash

fastboot devices
fastboot flash gpt PGPT
fastboot flash preloader preloader_amt6797_64_open.bin
fastboot flash recovery recovery.img
fastboot flash scp1 tinysys-scp.bin
fastboot flash scp2 tinysys-scp.bin
fastboot flash lk lk.bin
fastboot flash lk2 lk.bin
fastboot flash boot boot.img
fastboot flash logo logo.bin
fastboot flash tee1 trustzone.bin
fastboot flash tee2 trustzone.bin
fastboot flash system system.img
```

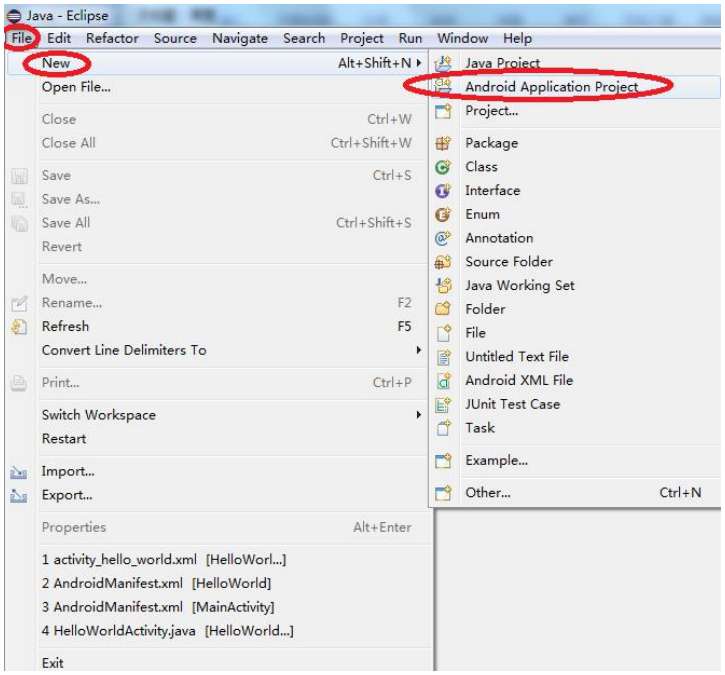
```
fastboot flash cache cache.img
fastboot flash userdata userdata.img
fastboot reboot
```

- Run the download script, download success.

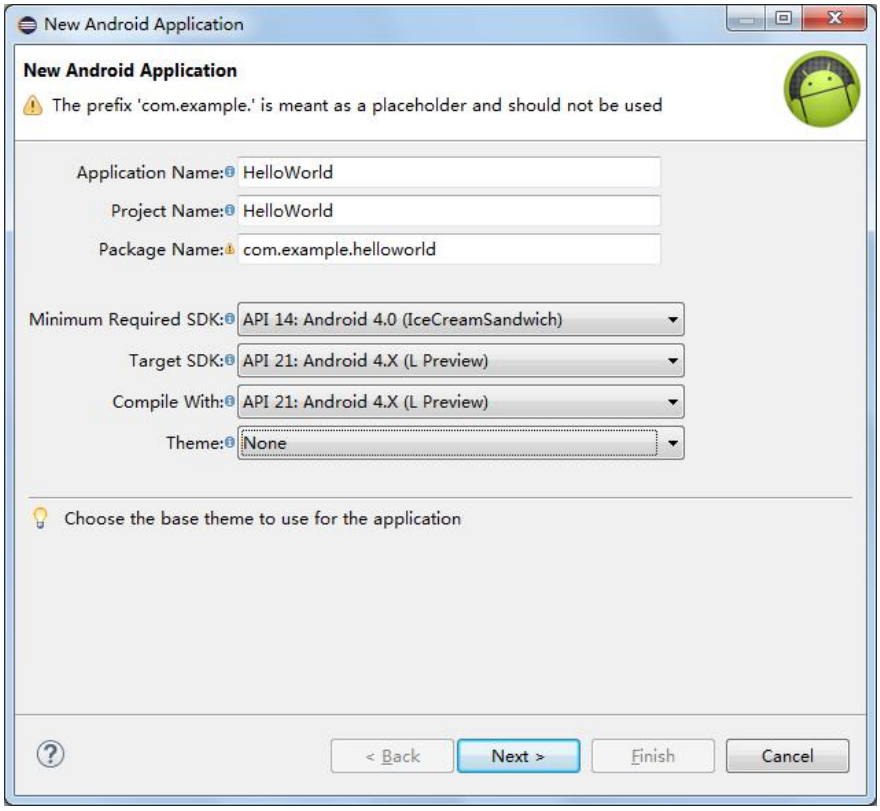
4. Android application demo

4.1. Create an android project

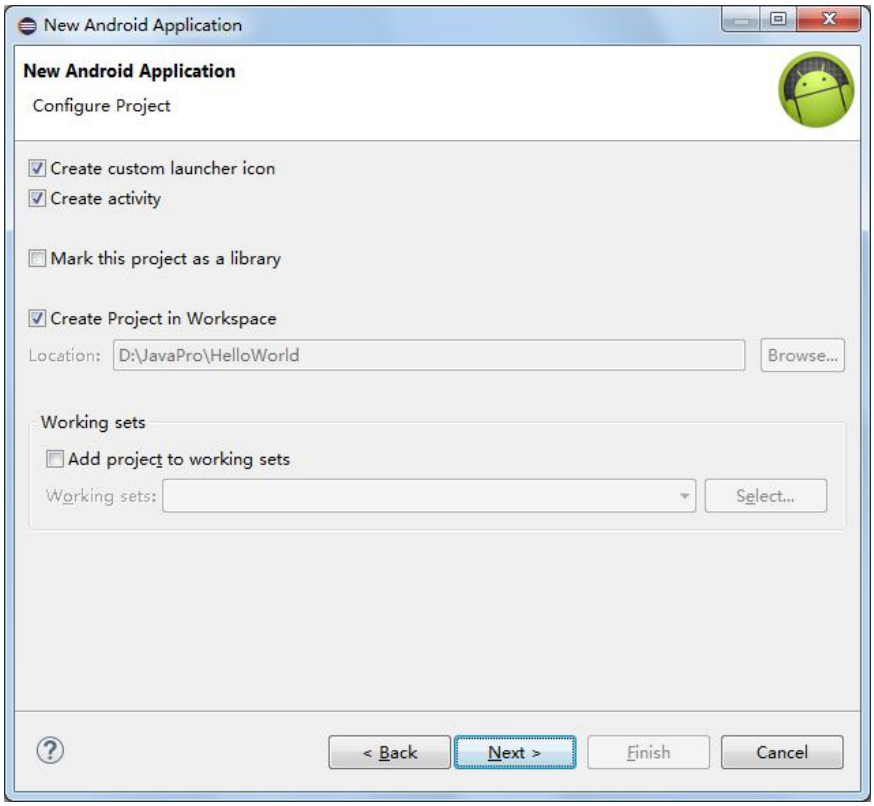
Step1. File-->New-->Android Application Project



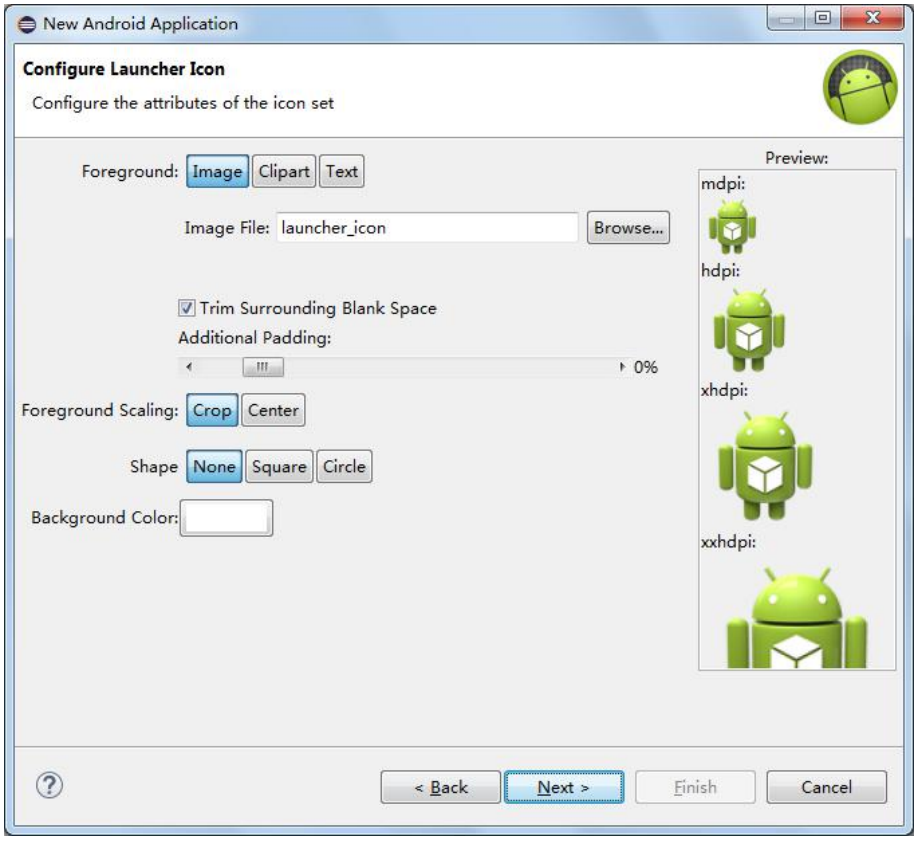
Step2. Input Application Name、Project Name、Package Name、choose SDK version、UI style,then click Next.



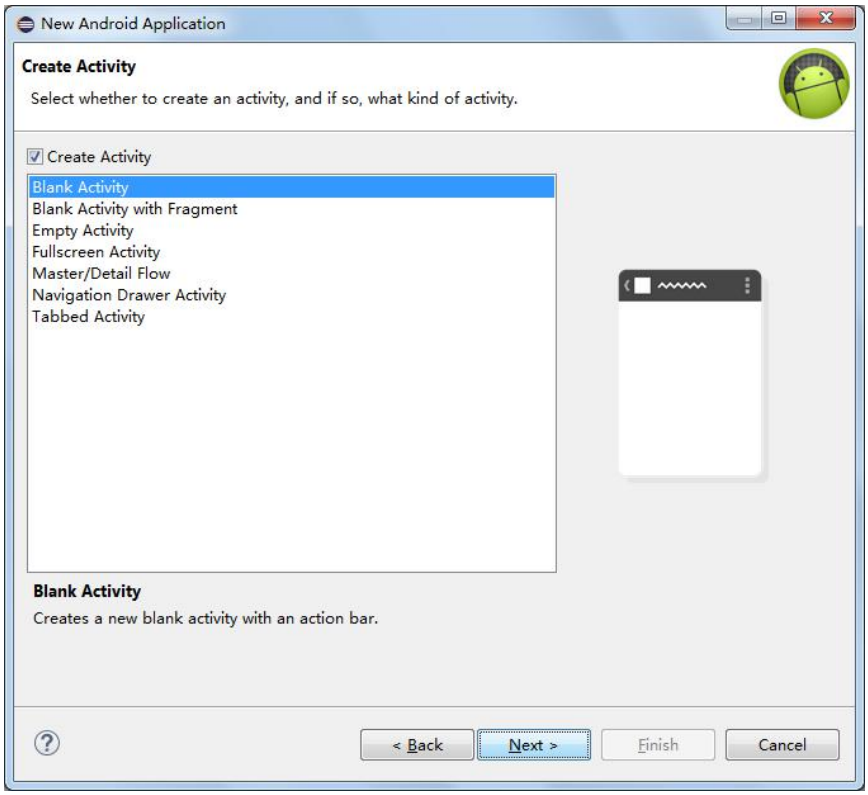
Step 3. Keep the default options, then click Next.



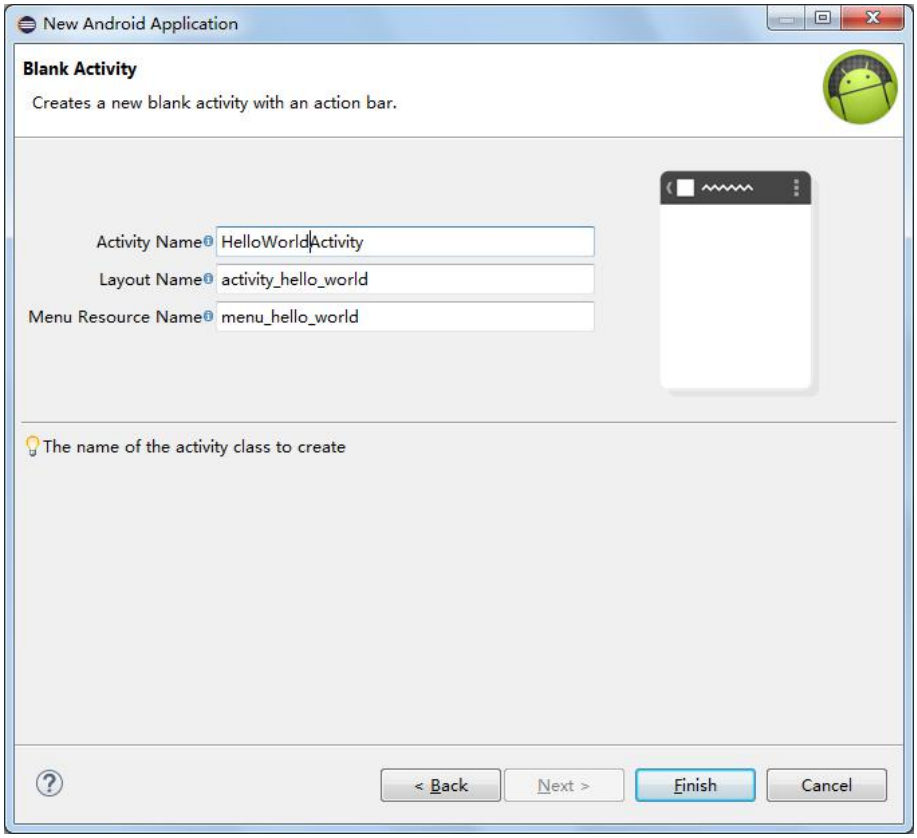
Step 4. Create the application start-up UI, then click Next.



Step 5. Create application activity template, select Blank Activity, then click Next.

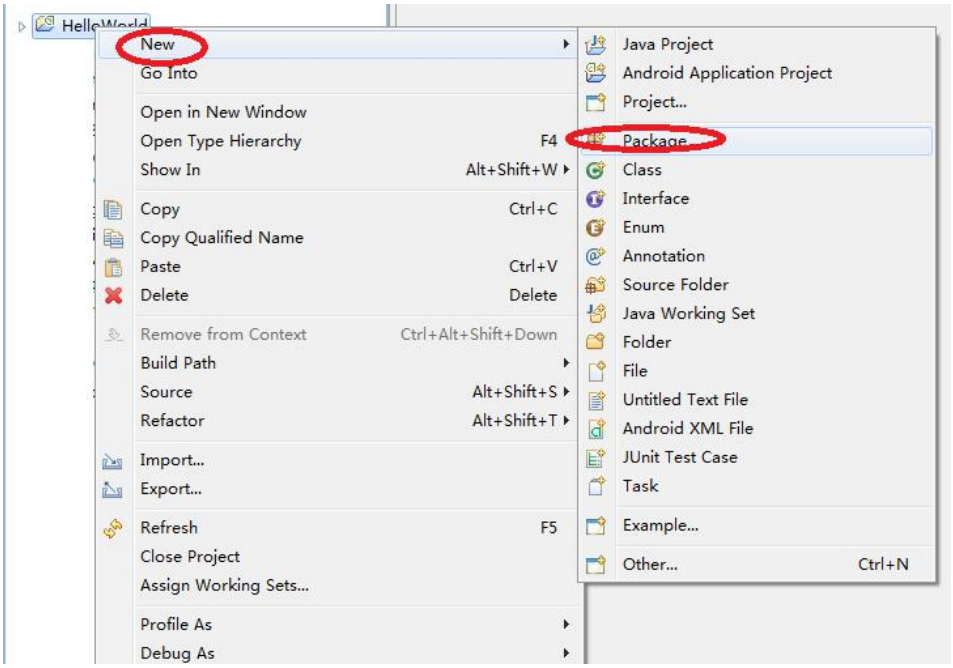


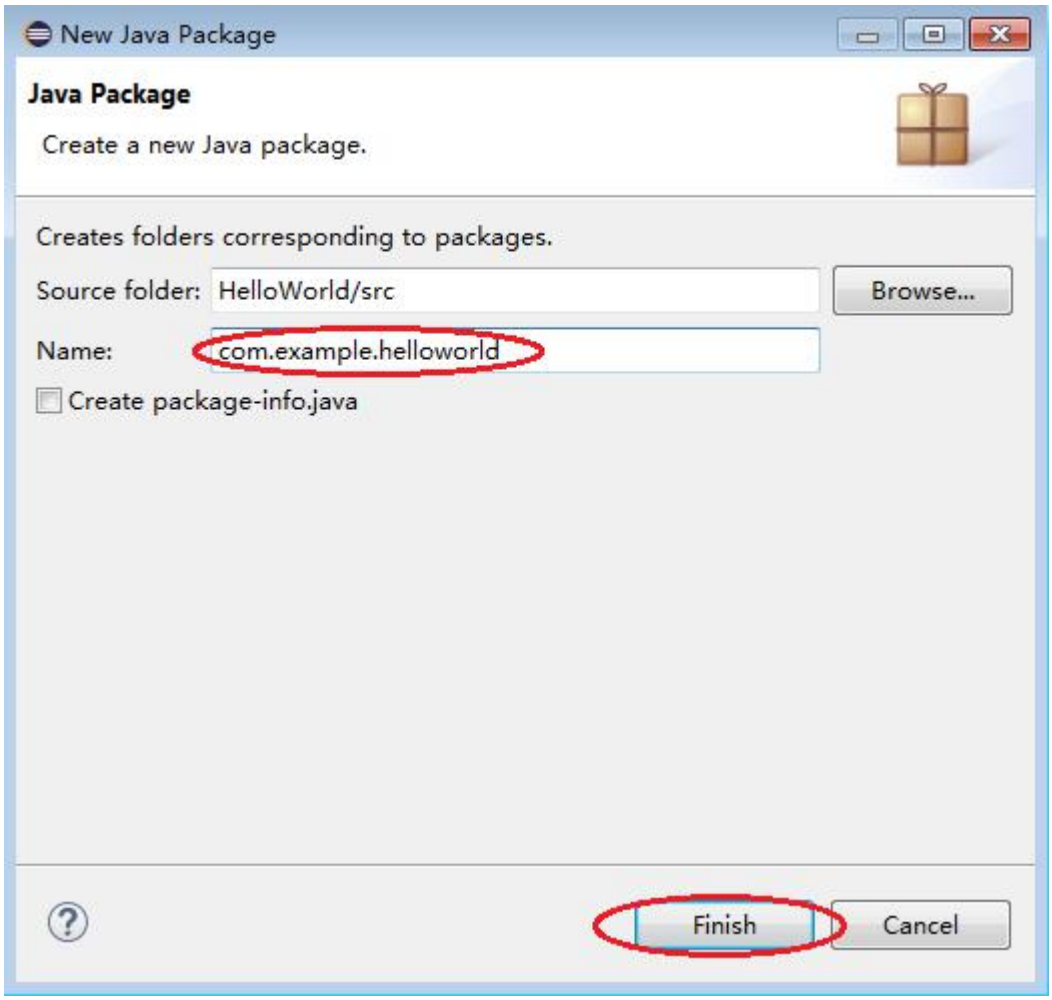
Step 6. Keep the default options, then click Finish to complete.



4.2. Create package

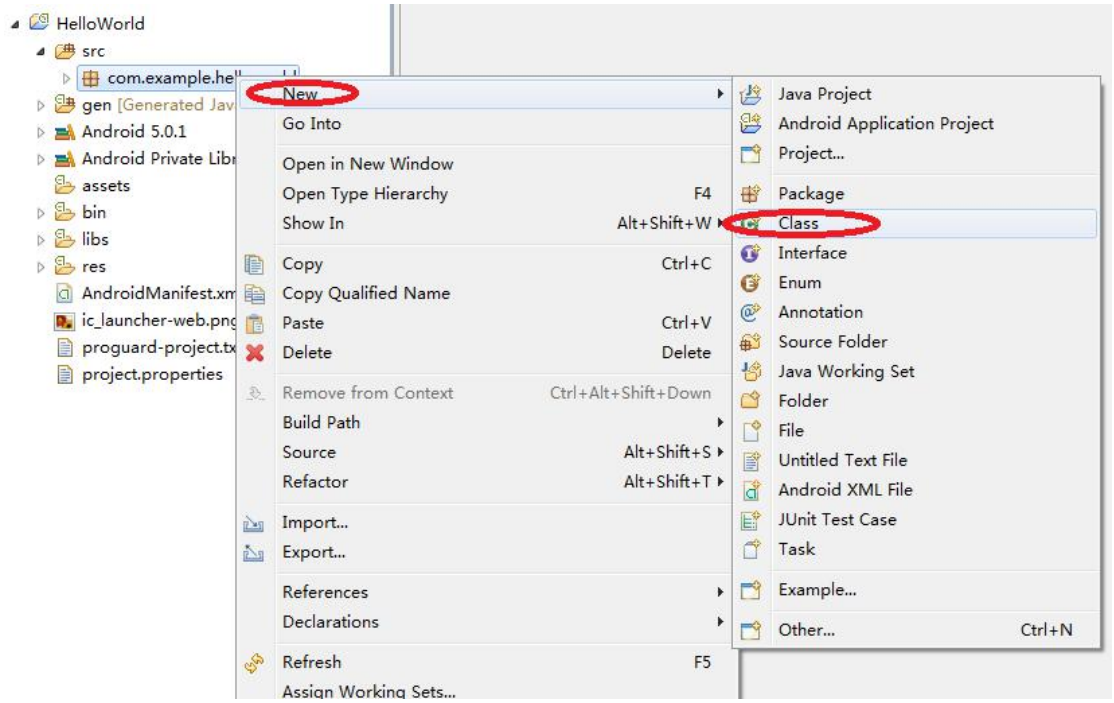
Right-click on the project name, select New -> Package, fill in the package name then click Finish to complete.

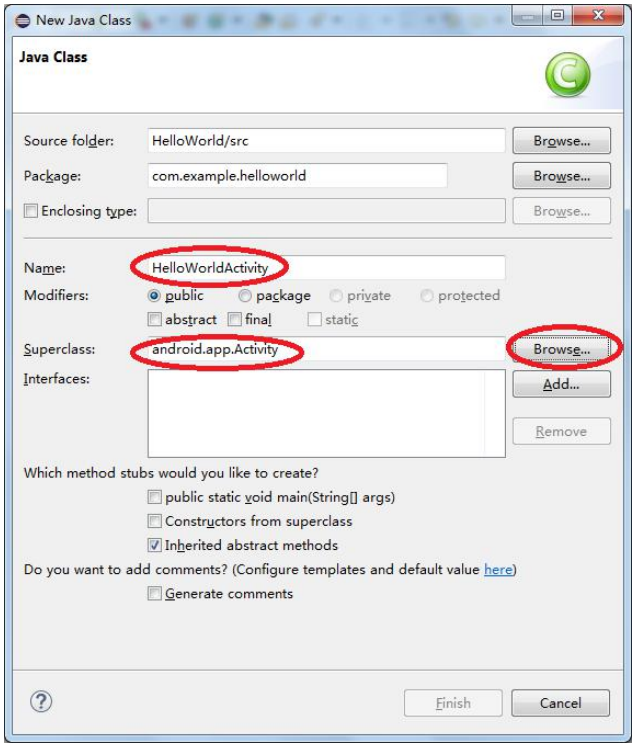




4.3. Create class

Right-click on the package name, select New -> Class, fill in the class name and choose the super class, then click Finish to complete.

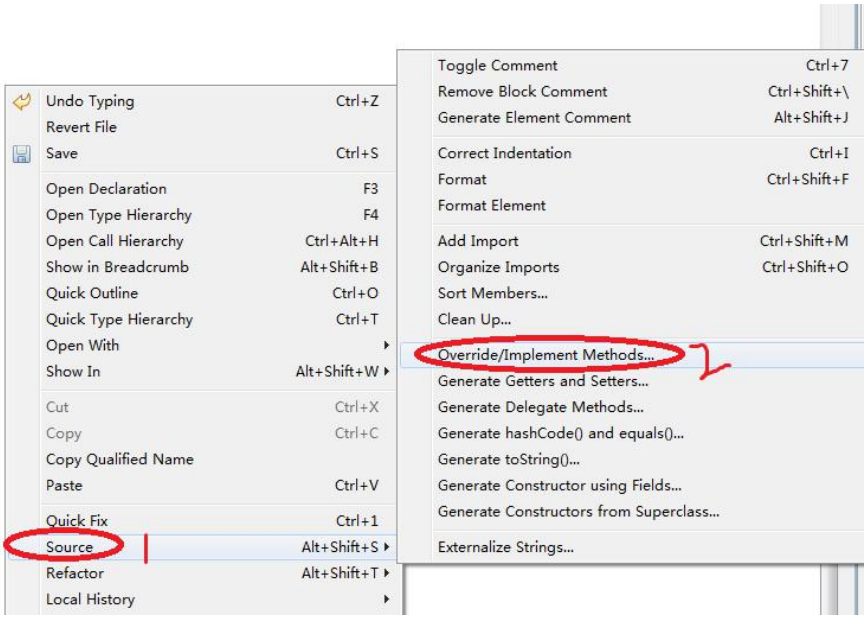

HelloWorld
src
com.example.hel
gen [Generated Jav
Android 5.0.1
Android Private Libr
assets
bin
libs
res
AndroidManifest.xm
ic_launcher-web.png
proguard-project.tx
project.properties
New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
References
Declarations
Refresh F5
Assign Working Sets...
Java Project
Android Application Project
Project...
Package
Class
Interface
Enum
Annotation
Source Folder
Java Working Set
Folder
File
Untitled Text File
Android XML File
JUnit Test Case
Task
Example...
Other... Ctrl+N

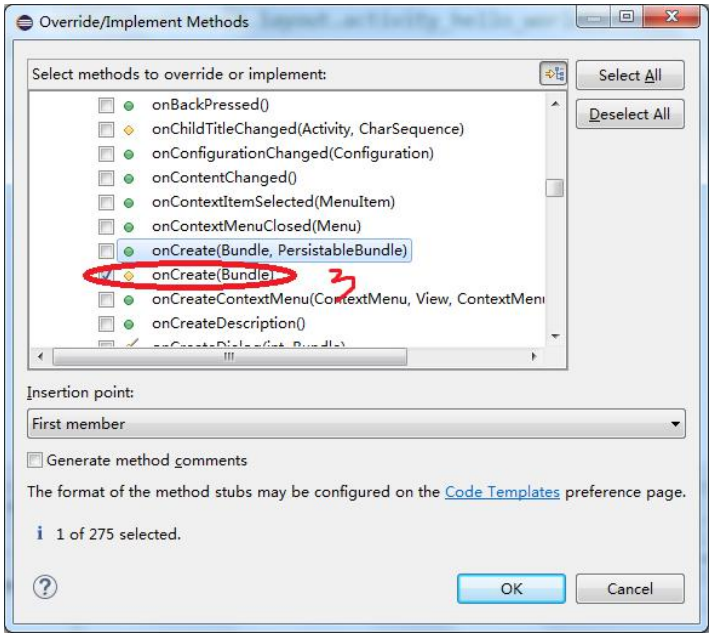

New Java Class
Java Class
Source folder: HelloWorld/src Browse...
Package: com.example.helloworld Browse...
Enclosing type: Browse...
Name: HelloWorldActivity
Modifiers: public package private protected
abstract final static
Superclass: android.app.Activity Browse...
Interfaces: Add... Remove
Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
Generate comments
Finish Cancel

HelloWorldActivity.java

```
1  package com.example.helloworld;
2
3  import android.app.Activity;
5
6  public class HelloWorldActivity extends Activity
7  {
8
9  }
10
11
12
```

Open HelloWorldActivity.java,Right-click,select Source -> Override/Implement Methods,you could override or implement the methods, and also you could implement the customization methods.



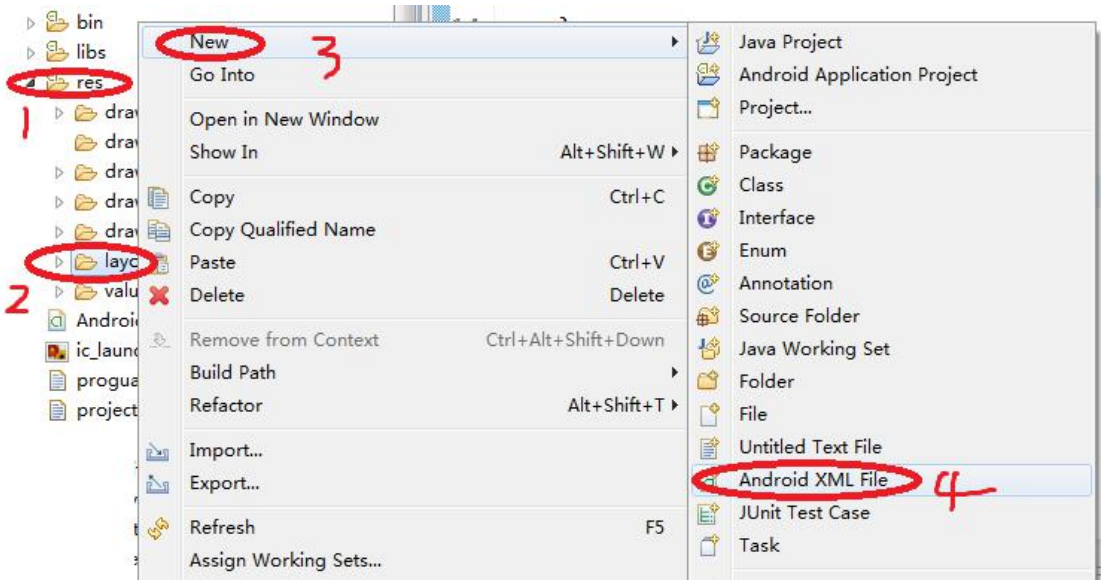


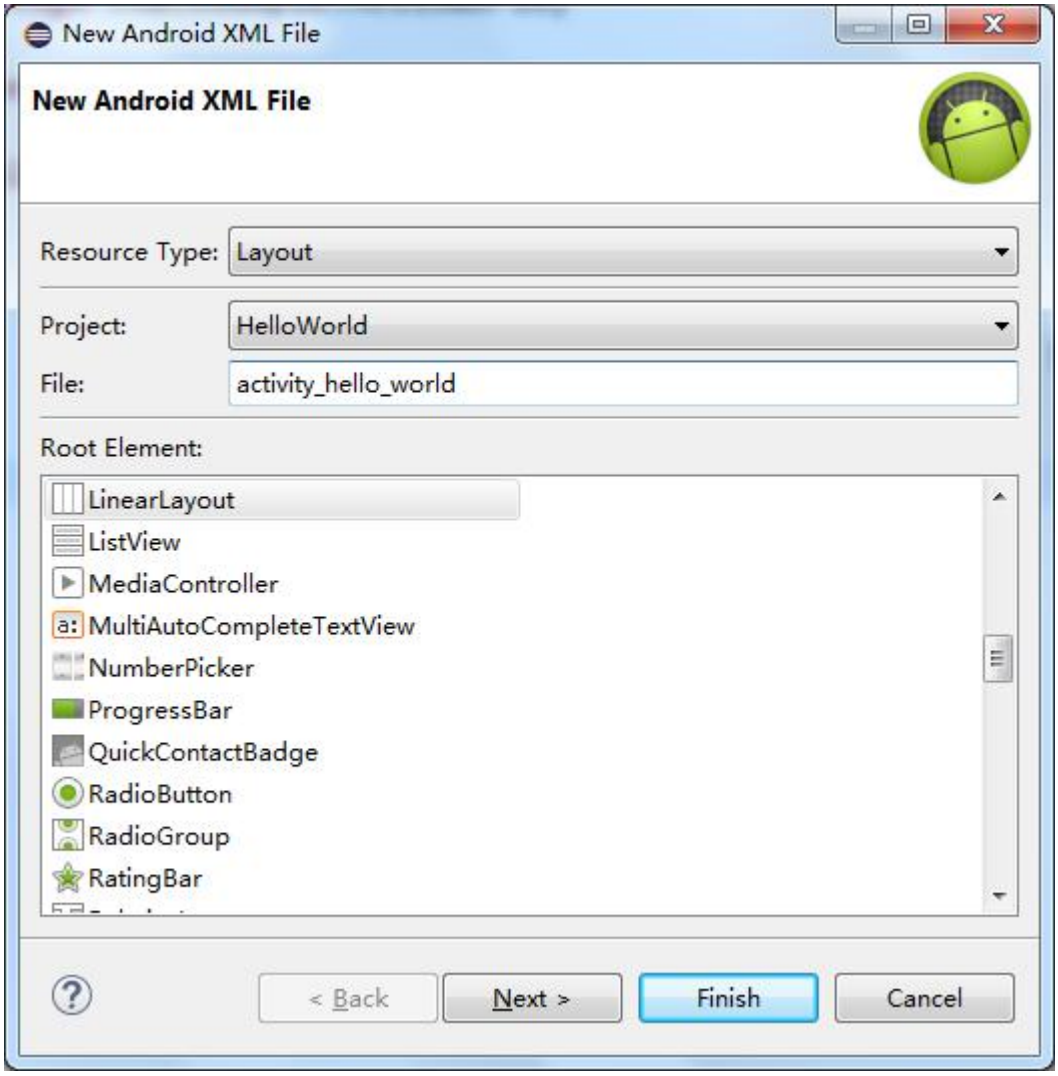
*HelloWorldActivity.java

```
1  package com.example.helloworld;
2
3  import android.app.Activity;
5
6  public class HelloWorldActivity extends Activity
7  {
8
9      @Override
10     protected void onCreate(Bundle savedInstanceState) {
11         // TODO Auto-generated method stub
12         super.onCreate(savedInstanceState);
13
14     }
15
16 }
```

4.4. Create layout file

Create a new XML file, and add a TextView then loads it into the activity.


bin
libs
res
1
2
New
3
Go Into
Open in New Window
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Assign Working Sets...
Java Project
Android Application Project
Project...
Package
Class
Interface
Enum
Annotation
Source Folder
Java Working Set
Folder
File
Untitled Text File
Android XML File
4
JUnit Test Case
Task


New Android XML File
New Android XML File
Resource Type: Layout
Project: HelloWorld
File: activity_hello_world
Root Element:
LinearLayout
ListView
MediaController
MultiAutoCompleteTextView
NumberPicker
ProgressBar
QuickContactBadge
RadioButton
RadioGroup
RatingBar
< Back
Next >
Finish
Cancel

HelloWorldActivity.java activity_hello_world.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="HelloWorld" />

</LinearLayout>
```

```java
public class HelloWorldActivity extends Activity
{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hello_world);
    }

}
```

4.5. Run the application

Configure the AndroidManifest.xml file, then run the application.

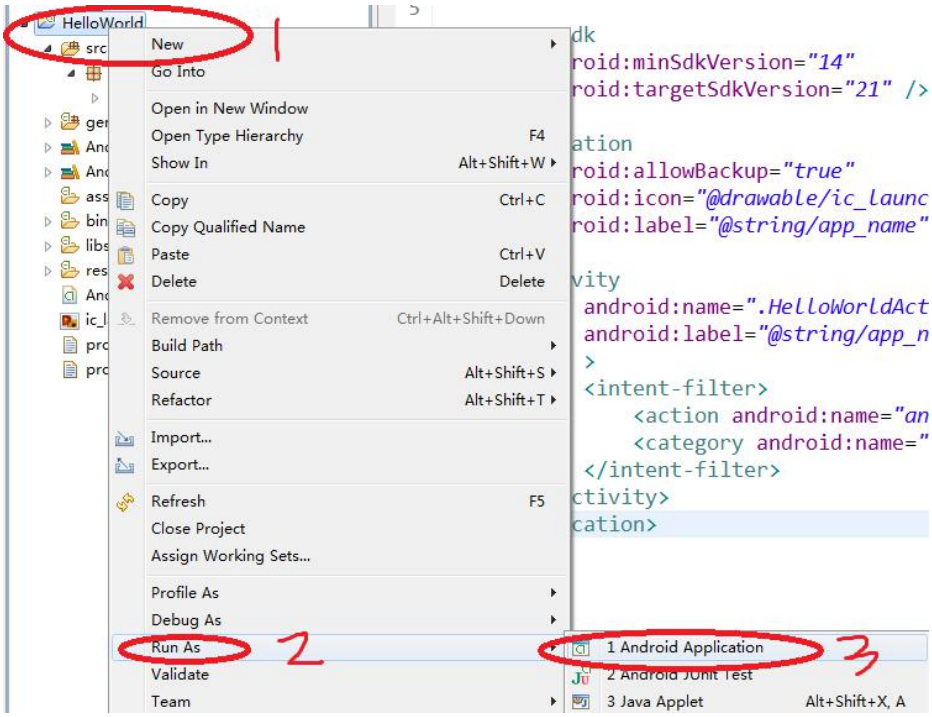
HelloWorldActivity.java | activity_hello_world.xml | HelloWorld Manifest

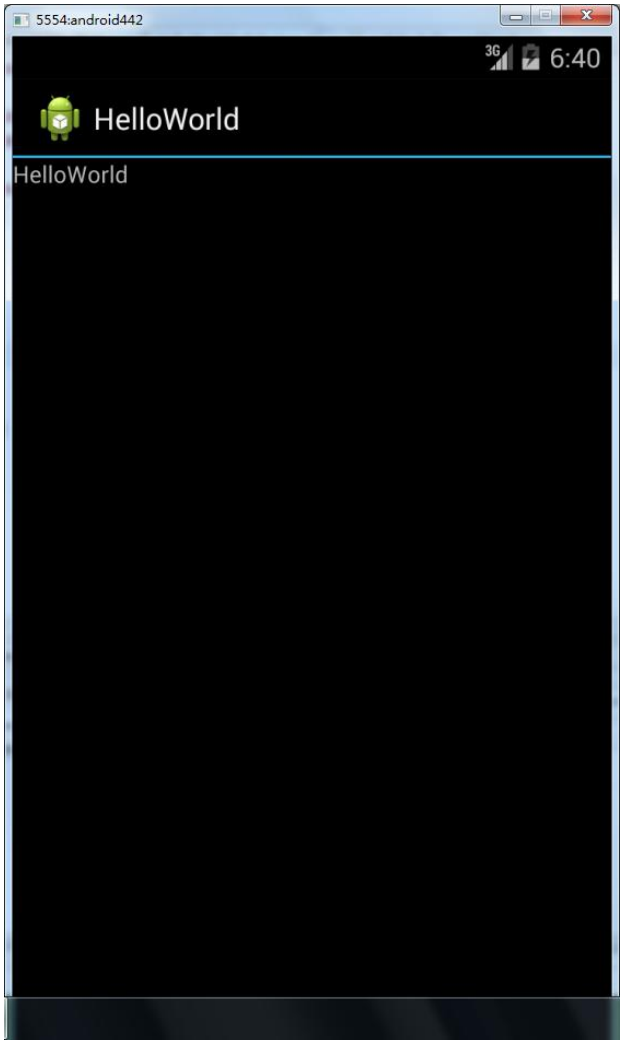
```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.helloworld"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="21" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        >
        <activity
            android:name=".HelloWorldActivity"
            android:label="@string/app_name"
            >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```




5554:android442
6:40
HelloWorld
HelloWorld

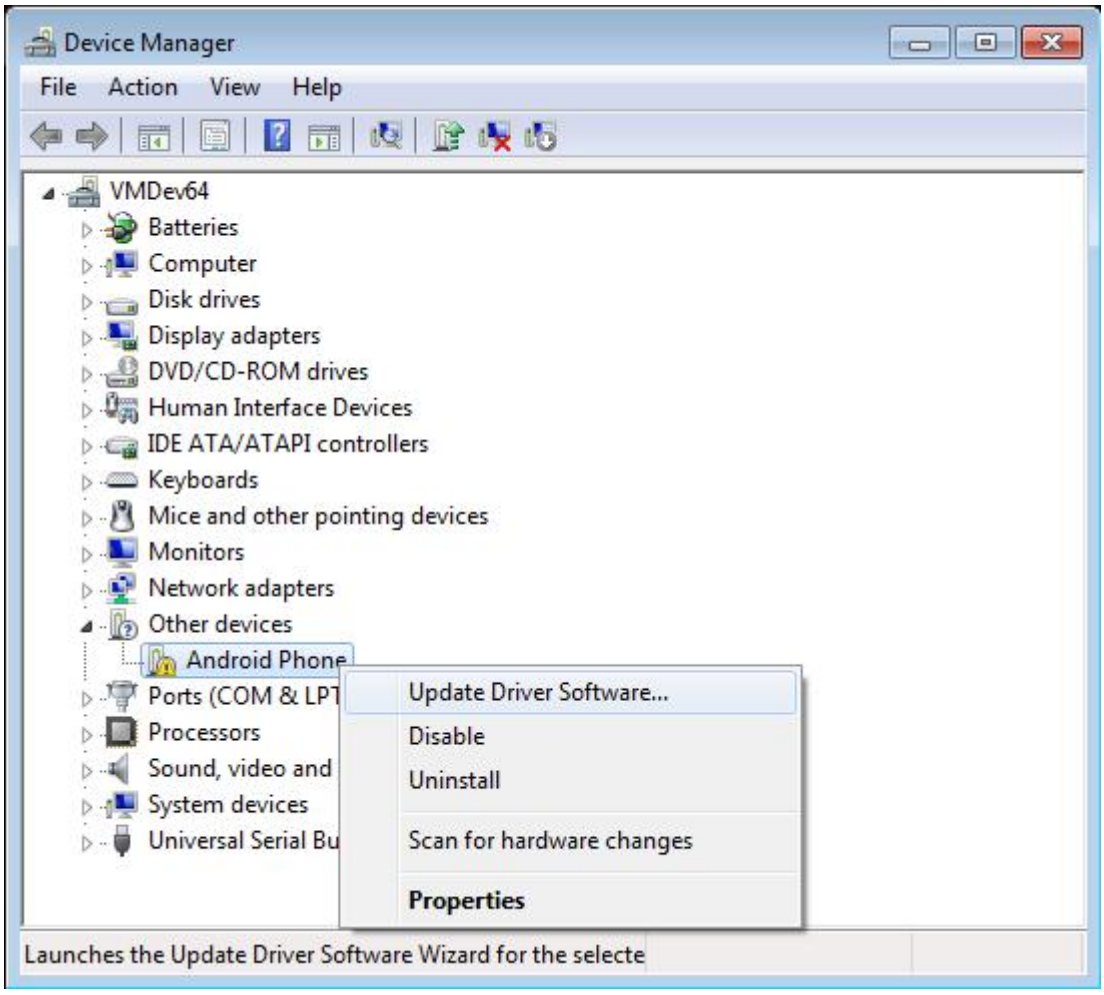
5. Android Debug Bridge

5.1. ADB introduction

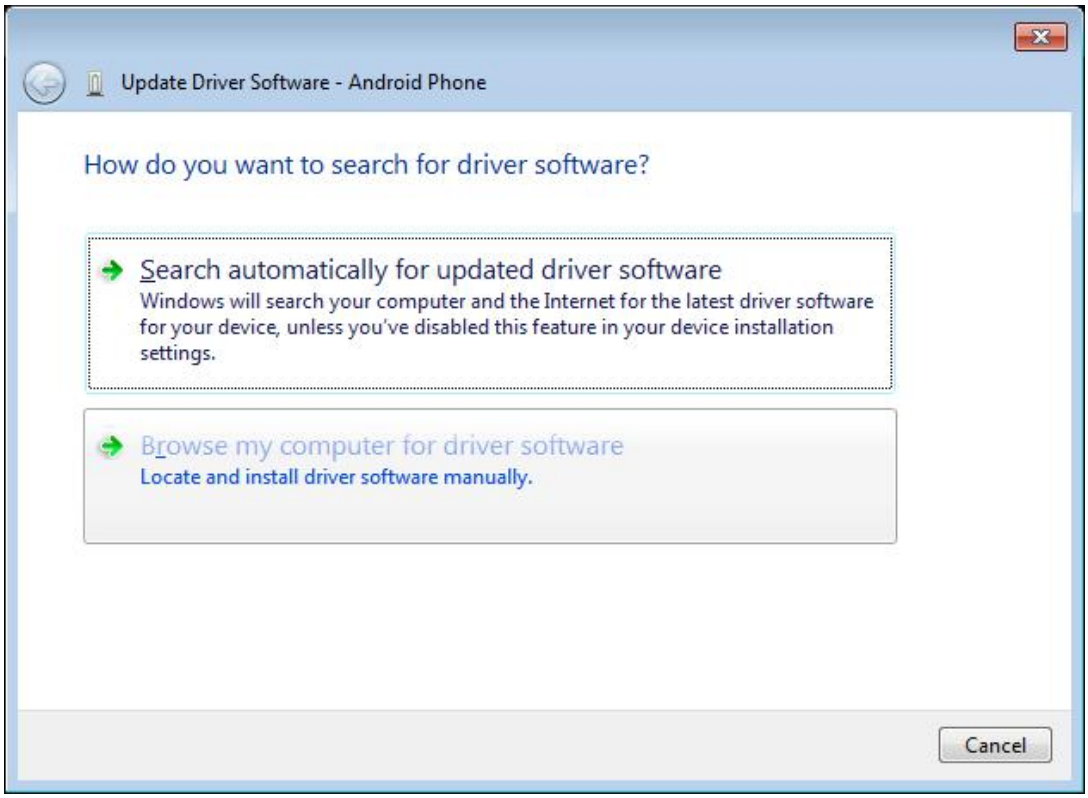
ADB, the full name is Android Debug Bridge, is an android command-line debug tool. It has a variety of functions, such as track system logs, upload and download files, install and uninstall applications.

5.2. ADB driver installation

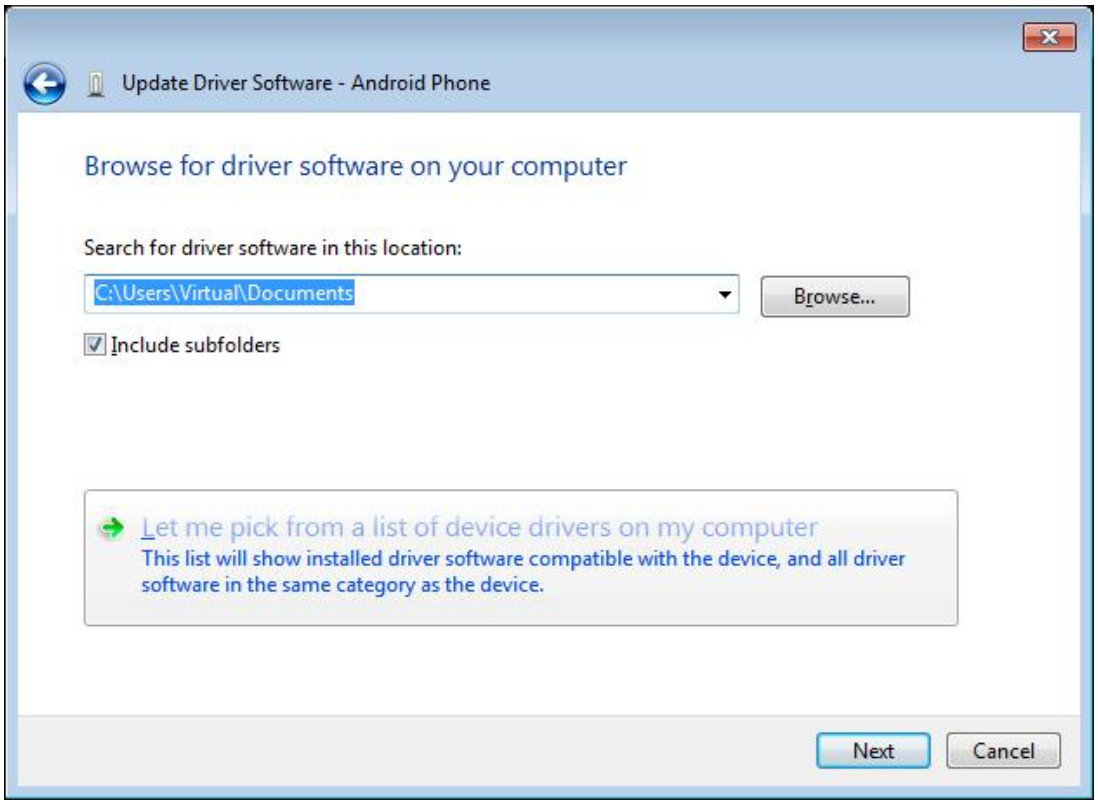
Step 1. Confirm the Google USB driver is installed, then plug in your device.
Warning: The driver won't install automatically. We will do it manually in the next steps.
Step 2. Open Device Manager and select your android device, right-click it and select “Update Driver Software”.



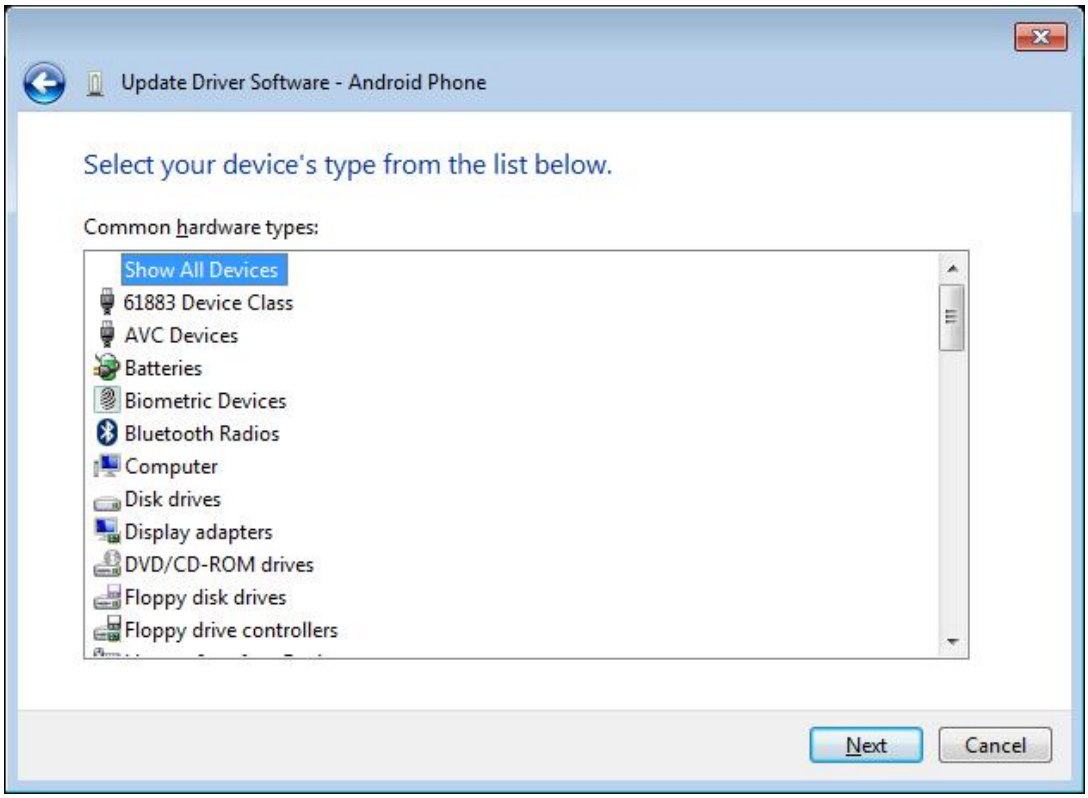
Step 3. Select "Browse my computer for driver software".



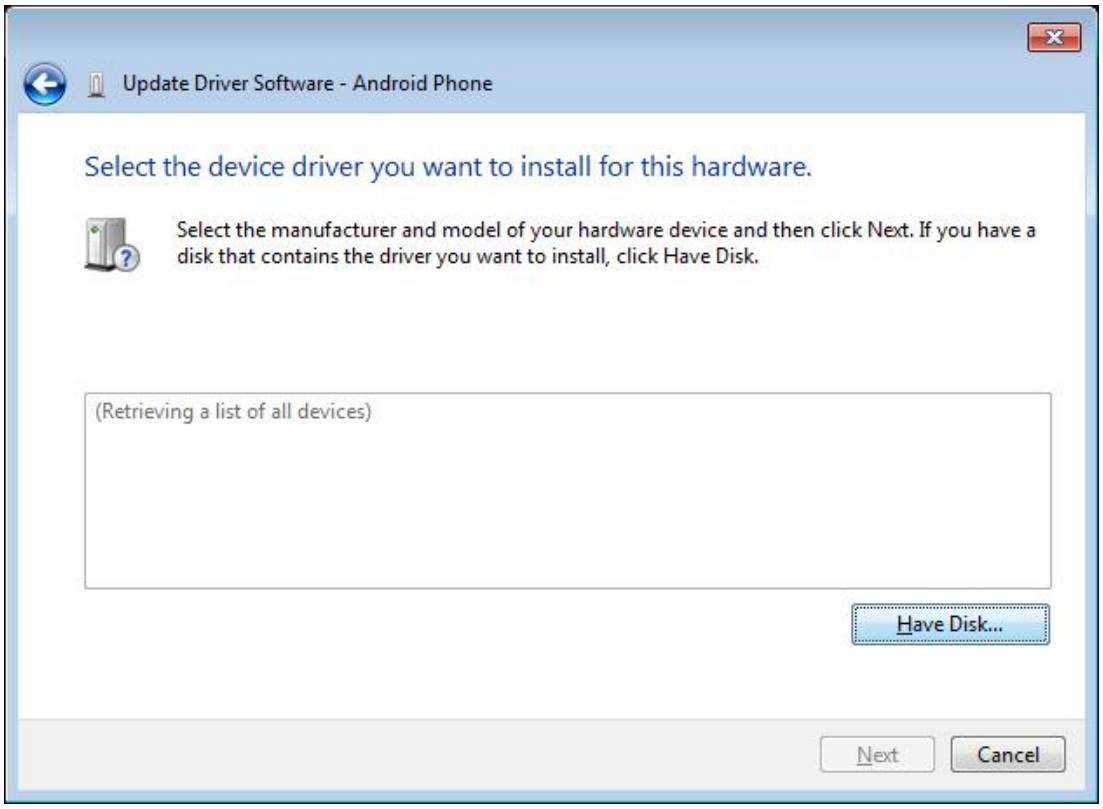
Step 4. Select "Let me pick from a list of device drivers on my computer".



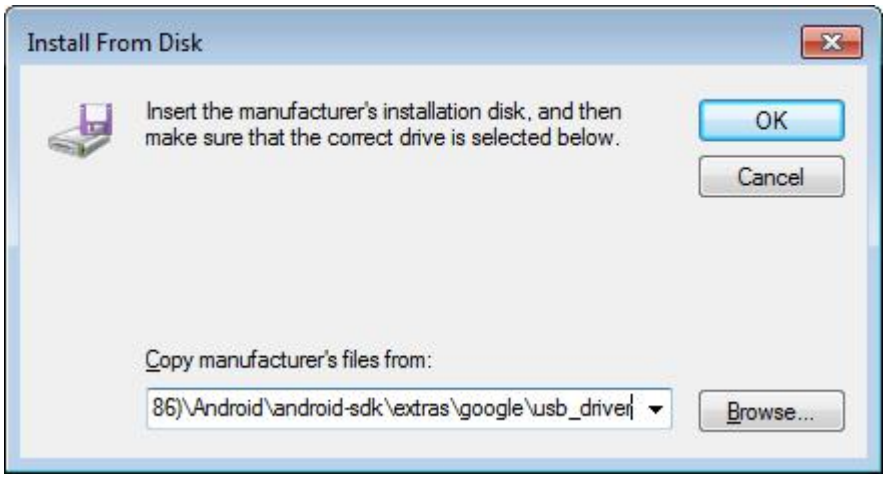
Step 5. Select "Show All Devices".



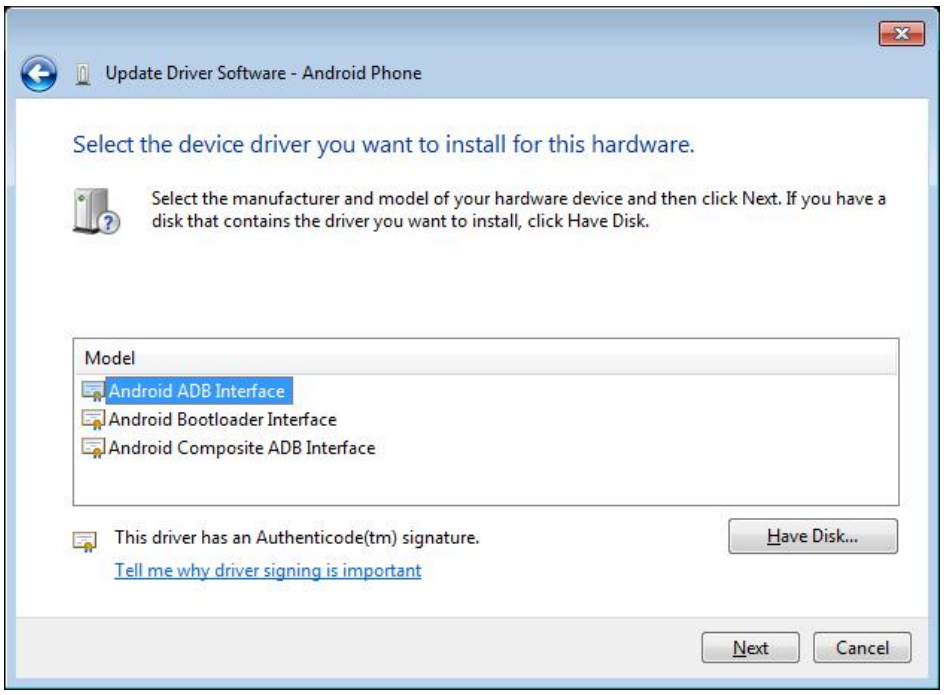
Step 6. Press the "Have Disk" button.



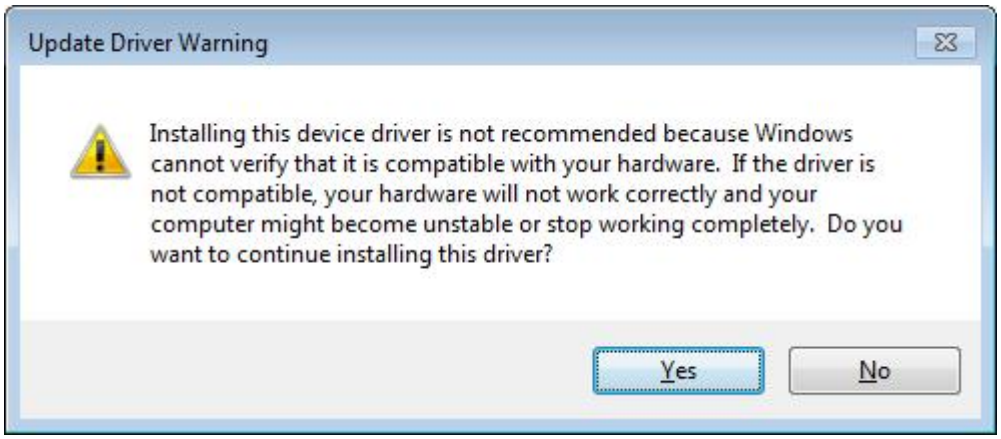
Step 7. Enter the path to the Google USB driver. Normally it is located in the following directory: C:\Program Files (x86)\Android\android-sdk\extras\google\usb_driver



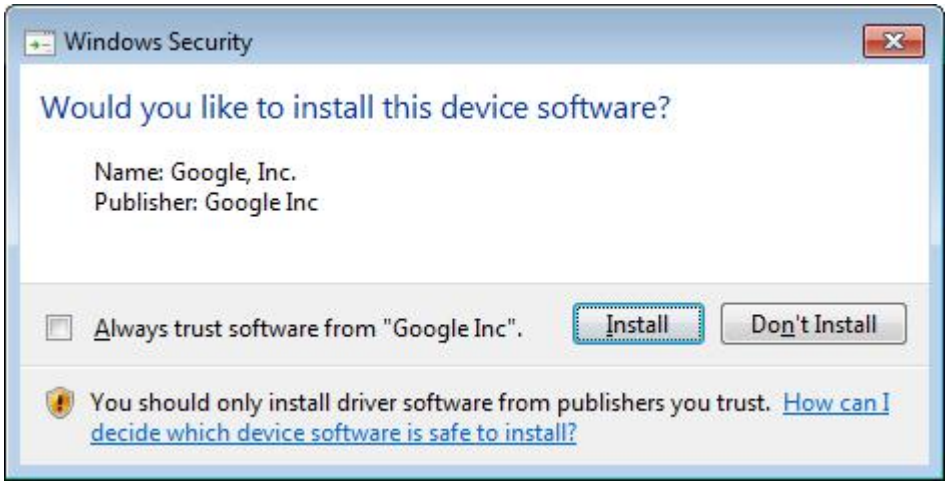
Step 8. Select "Android ADB Interface" from the list of device types.



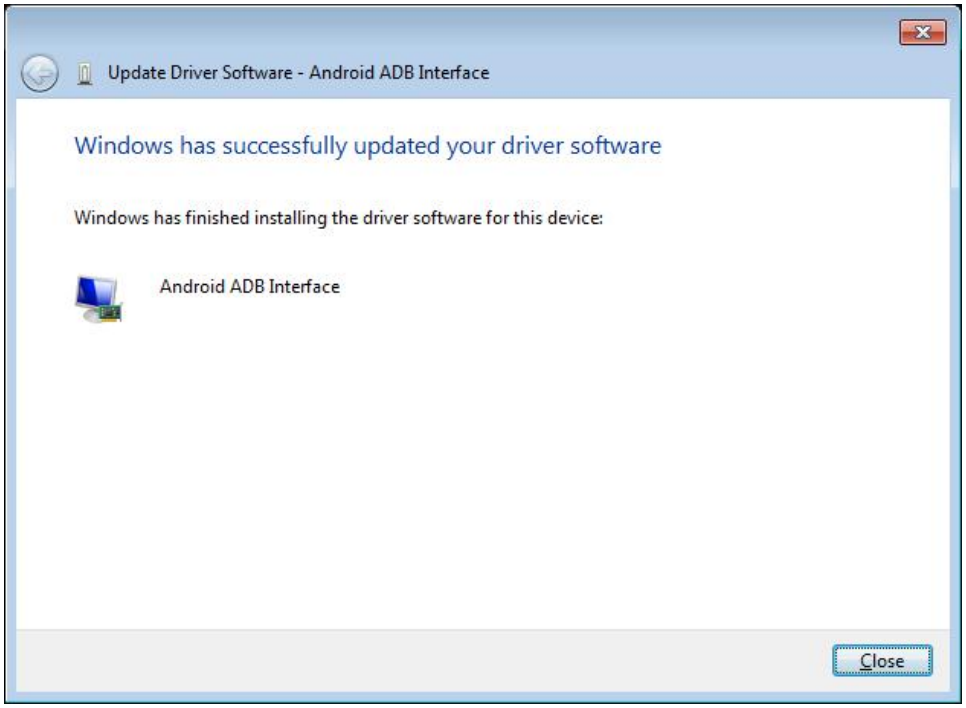
Step 9. Confirm the installation of the driver by pressing "Yes".



Step 10. Confirm the installation again by pressing "Install".

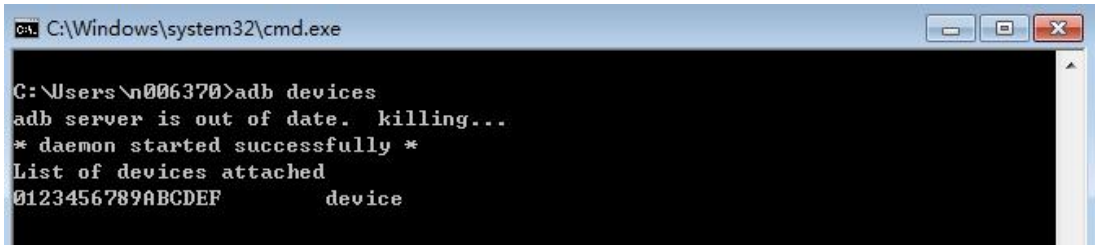


Step 11. When the installation is done, press "Close".



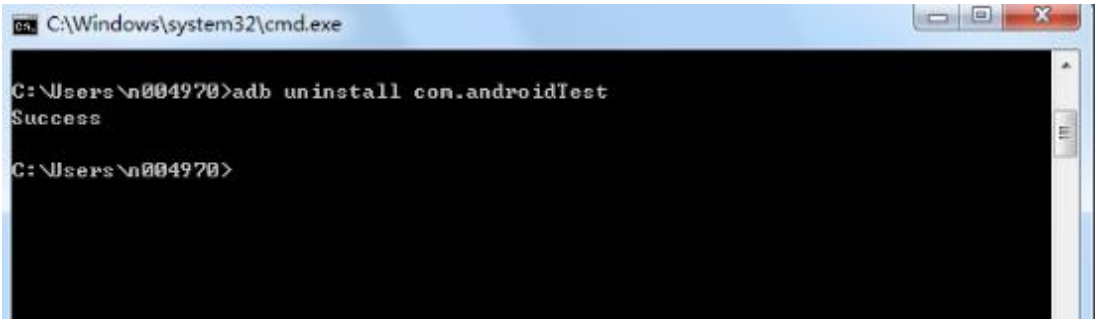
5.3. ADB commands

adb devices: view connection device command

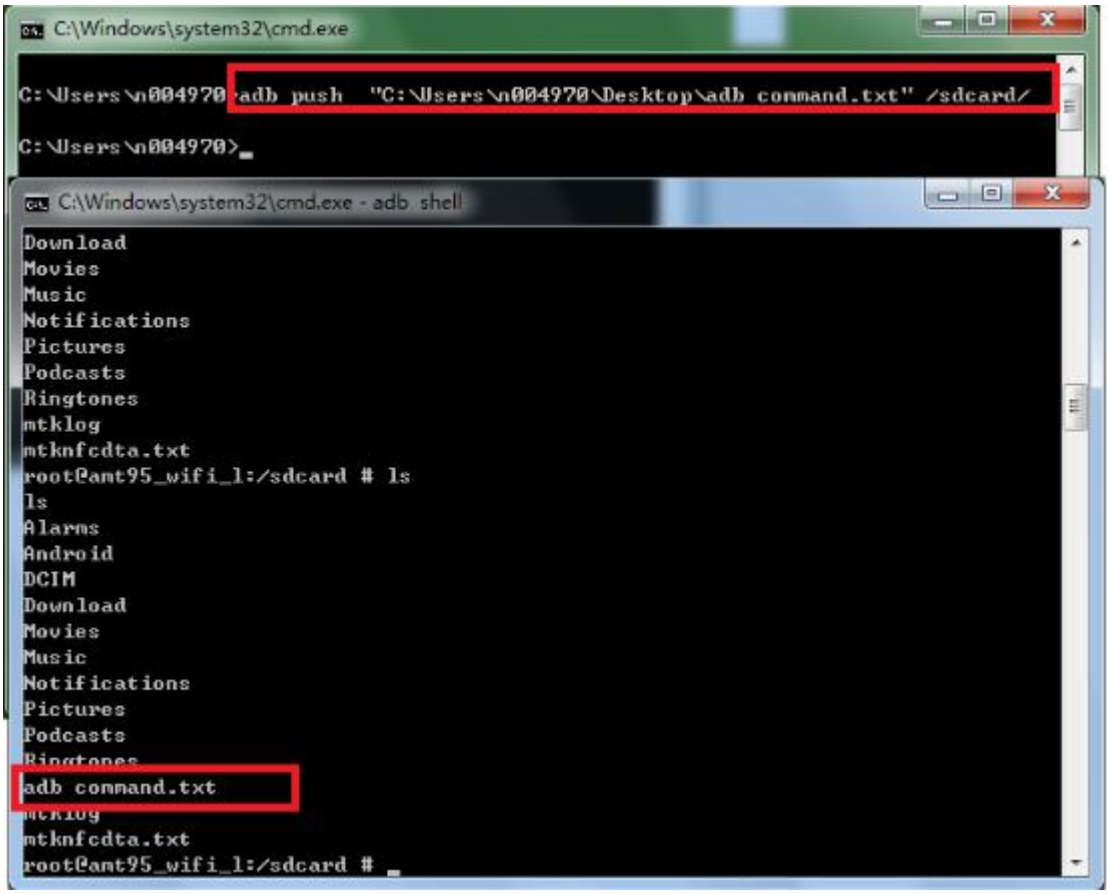


adb install: install apk command

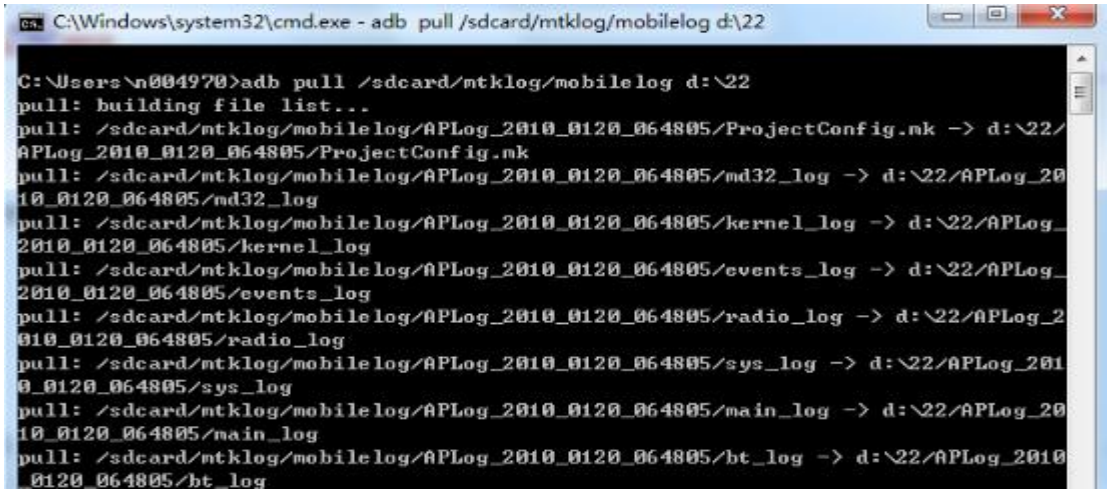
adb uninstall:uninstall apk command



adb push <local path> <remote path>:send files to the device from the computer

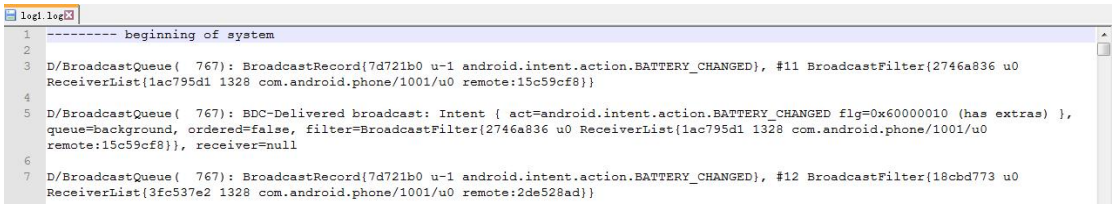


adb pull <remote path> <local path>:download files from the device to your computer

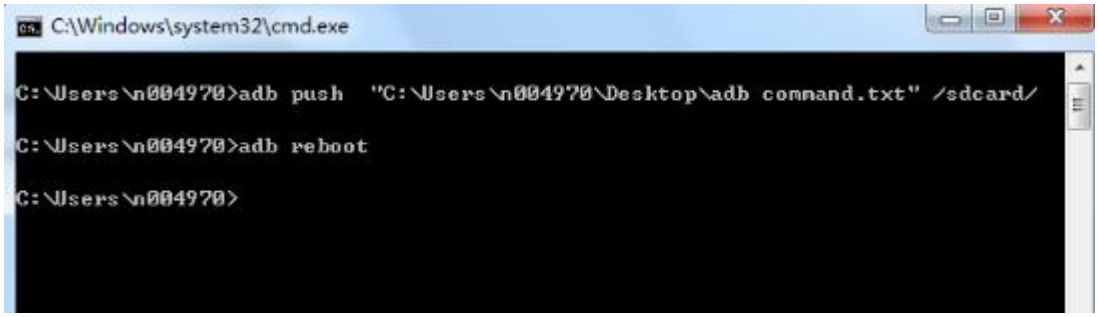


adb shell:enter the shell model of the device

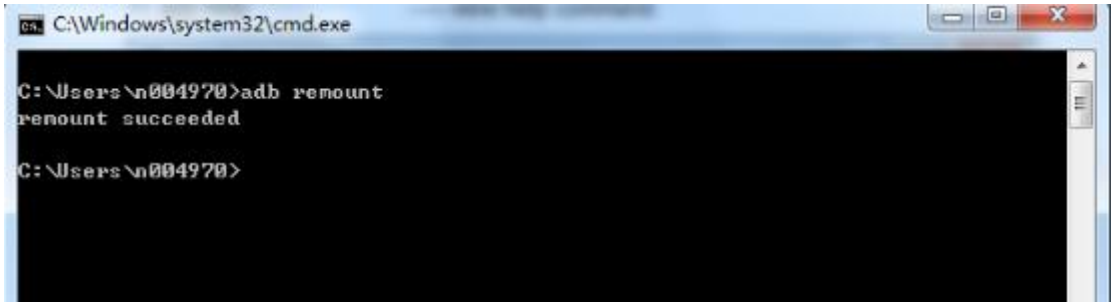
adb logcat:display log information



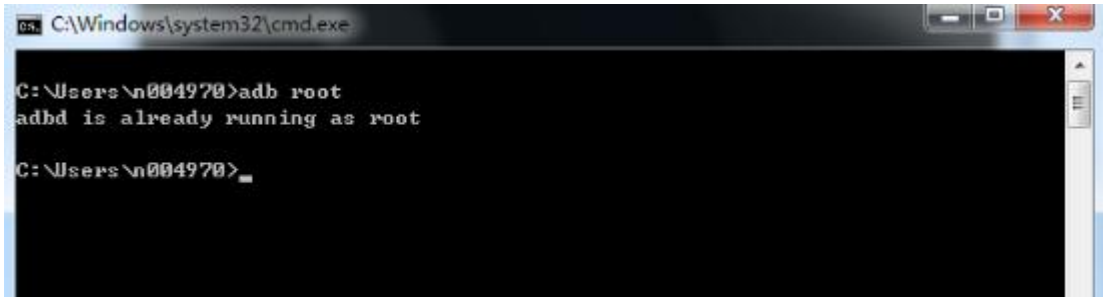
adb reboot:restart device



adb remount:remount the system partition and make it be able to write



adb root:get root privileges



adb help:display help information