## adb 常用命令

（如是多个终端 adb -s 终端（例如198.9.9.202:5555）...）

1，copy 文件到终端

adb push pc目录 终端目录

2, copy 文件到电脑

adb pull 终端目录 pc目录

3，查看终端目录

adb shell

ls

4,查看文件路径

adb shell

ls

pwd

5,打开一个activity

adb shell am start -n 包名/类名

adb -s 198.9.9.202:5555 shell am start -n com.android.settings/.Settings

6,打开一个service

adb shell am startsevice -n 包名类名

adb -s 198.9.9.202:5555 shell am startservice -n com.bubblesoft.android.bubbleupnp/.AndroidUpnpService

7,查看应用签名信息

jarsigner -verify -verbose -certs 路径+包名

## 页面全屏

getWindow().setFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN, WindowManager.LayoutParams.FLAG\_FULLSCREEN);

//隐去电池等图标和一切修饰部分（状态栏部分）

requestWindowFeature(Window.FEATURE\_NO\_TITLE);

// 隐去标题栏（程序的名字）

手动全屏

getWindow().addFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN);

取消全屏

getWindow().clearFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN);

## eclipse 无法启动

删除 .metadata\.plugins\org.eclipse.core.resources 目录下.snap 文件

## EditText 限制输入内容

android:digits="0123456789."

## Dialog 点击按扭不消失

try {

//java.lang.reflect.Field;

Field field = dialog.getClass()

.getSuperclass()

.getDeclaredField("mShowing");

field.setAccessible(true);

field.set(dialog, false);//true消失，false 不消失

} catch (Exception e) {

e.printStackTrace();

}

## 初始化屏幕尺寸

@Override

**public** **void** onWindowFocusChanged(**boolean** hasFocus) {

// **TODO** Auto-generated method stub

**super**.onWindowFocusChanged(hasFocus);

initscreensize();

}

**private** **void** initscreensize() {

// **TODO** Auto-generated method stub

*screenWidth* = getWindow().getWindowManager().getDefaultDisplay()

.getWidth();

*screenHeight* = getWindow().getWindowManager().getDefaultDisplay()

.getHeight();

Rect frame = **new** Rect();

getWindow().getDecorView().getWindowVisibleDisplayFrame(frame);

// 状态栏高度

**int** statusBarHeight = frame.top;

// int contentTop = getWindow().findViewById(Window.ID\_ANDROID\_CONTENT)

// .getTop();

// // statusBarHeight是上面所求的状态栏的高度

// int titleBarHeight = contentTop - statusBarHeight;

*screenHeight* -= statusBarHeight;

**if** (gallery != **null**) {

gallery.setscreensize(*screenWidth*, *screenHeight*);

}

}

## 获取屏幕密度

4.密度Density

根据像素分辨率，在屏幕指定物理宽高范围内能显示的像素数量。

在同样的宽高区域，低密度的显示屏能显示的像素较少，而高密度的显示屏则能显示更多的像素。

屏幕密度非常重要，因为其它条件不变的情况下，一共宽高固定的UI组件（比如一个按钮）在在低密度的显示屏上显得很大， 而在高密度显示屏上看起来就很小。

为简单起见，Android把所有的屏幕分辨率也分为四种尺寸：小，普通，大，超大(分别对应：small, normal, large, and extra large).

应用程序可以为这四种尺寸分别提供不同的资源-平台将透明的对资源进行缩放以适配指定的屏幕分辨率。

密度无关的像素（ DIP ）   
指一个抽象意义上的像素，程序用它来定义界面元素。它作为一个与实际密度无关的单位，帮助程序员构建一个布局方案（界面元素的宽度，高度，位置）。   
一个与密度无关的像素，在逻辑尺寸上，与一个位于像素密度为 160DPI的屏幕上的像素是一致的，这也是Android平台所假定的默认显示设备。在运行的时候，平台会以目标屏幕的密度作为基准，“透明地”处理所 有需要的DIP缩放操作。要把密度无关像素转换为屏幕像素，可以用这样一个简单的公式： pixels = dips \* (density / 160)。举个例子，在 DPI为 240的屏幕上， 1个 DIP等于 1.5个物理像素。我们强烈推荐你用 DIP来定义你程序的界面布局，因为这样可以保证你的 UI在各种分辨率的屏幕上都可以正常显示

  在一个Activity的onCreate方法中，写入如下代码：

        DisplayMetrics metric = new DisplayMetrics();  
        getWindowManager().getDefaultDisplay().getMetrics(metric);  
        int width = metric.widthPixels;  // 屏幕宽度（像素）  
        int height = metric.heightPixels;  // 屏幕高度（像素）  
        float density = metric.density;  // 屏幕密度（0.75 / 1.0 / 1.5）  
        int densityDpi = metric.densityDpi;  // 屏幕密度DPI（120 / 160 / 240）

    但是，需要注意的是，在一个低密度的小屏手机上，仅靠上面的代码是不能获取正确的尺寸的。比如说，一部240x320像素的低密度手机，如果运行上述代码，获取到的屏幕尺寸是320x427。因此，研究之后发现，若没有设定多分辨率支持的话，Android系统会将240x320的低密度（120）尺寸转换为中等密度（160）对应的尺寸，这样的话就大大影响了程序的编码。所以，需要在工程的AndroidManifest.xml文件中，加入supports-screens节点，具体的内容如下：  
        <supports-screens  
            android:smallScreens="true"  
            android:normalScreens="true"  
            android:largeScreens="true"  
            android:resizeable="true"  
            android:anyDensity="true"/>

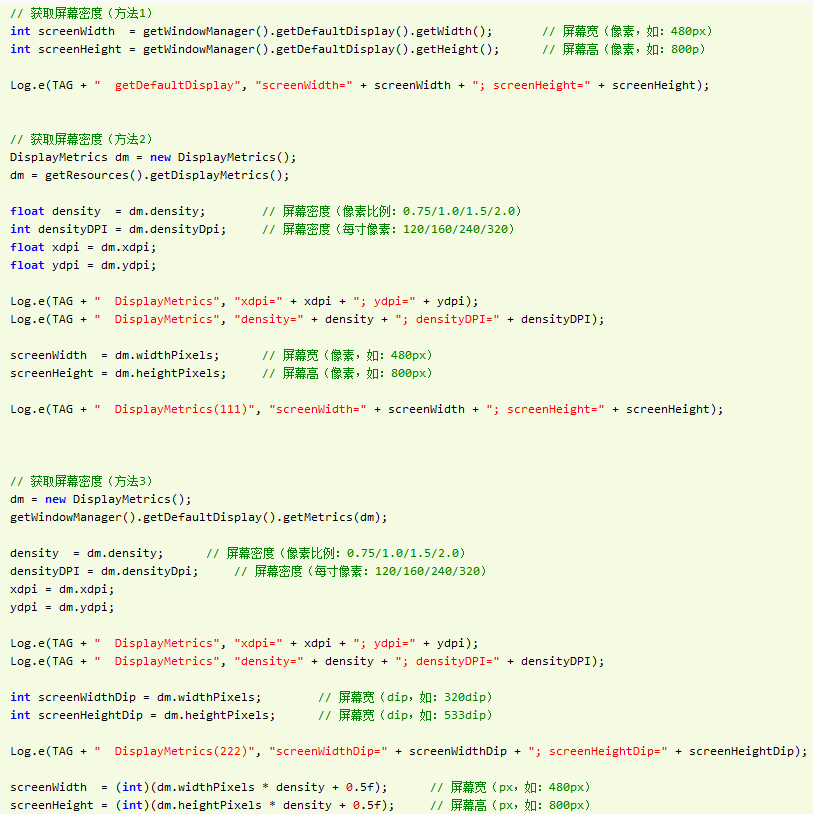
    这样的话，当前的Android程序就支持了多种分辨率，那么就可以得到正确的物理尺寸了。

// 获取屏幕密度（方法1）

int screenWidth  = getWindowManager().getDefaultDisplay().getWidth();       // 屏幕宽（像素，如：480px）

int screenHeight = getWindowManager().getDefaultDisplay().getHeight();      // 屏幕高（像素，如：800p）

Log.e(TAG + "  getDefaultDisplay", "screenWidth=" + screenWidth + "; screenHeight=" + screenHeight);



## 获取文字的高度和宽度

Rect bounds = new Rect();

String text = "Hello World";

TextPaint paint;

paint = findViewById(R.id.hello\_world).getPaint();

paint.getTextBounds(text, 0, text.length(), bounds);

int width = bounds.width();

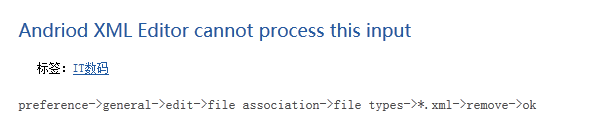
Paint pFont = new Paint();

Rect rect = new Rect();

pFont.getTextBounds("豆",0, 1, rect);

Log.v(TAG, "height:"+rect.height()+"width:"+rect.width());

## Android XML Editor cannot process this input



## **Android SDK Manager下载过慢的问题**

如果开启了goagent的话，这个问题其实很好解决，在打开的Android SDK Manager上方选择tools----Options，HTTP proxy server填写127.0.0.1, HTTP proxy port填写8087，关闭后重启eclipse，打开Android SDK Manager就会发现下载速度“杠杠”的～

从中其实可以看出，goagent的工作原理其实更像是把自己本机的电脑设置成了一个代理服务器，开启了8087的代理端口。

摘录自：http://www.cnblogs.com/kongkongyzt/p/3236155.html

感谢！

另一种方法也可以一试：能解决国内访问Google服务器的困难  
启动 Android SDK Manager ，打开主界面，依次选择「Tools」、「Options...」，弹出『Android SDK Manager - Settings』窗口；  
在『Android SDK Manager - Settings』窗口中，在「HTTP Proxy Server」和「HTTP Proxy Port」输入框内填入mirrors.neusoft.edu.cn和80，并且选中「Force https://...sources to be fetched using http://...」复选框。设置完成后单击「Close」按钮关闭『Android SDK Manager - Settings』窗口返回到主界面；  
依次选择「Packages」、「Reload」。

## SQL语句

### 1，关联查询

**public** List findLessonHomeWorkDetail1(String startdate, String enddate,

String classid) {

String queryString = "select " +

"ifnull(a.id,'') as cid," +

"a.class\_name," +

"a.class\_num," +

"ifnull(b.id,'') as sid," +

"b.student\_name," +

"ifnull(c.id,'') as ldid," +

"c.end\_time," +

"c.lesson\_address," +

"c.lesson\_desc," +

"c.lesson\_end," +

"c.lesson\_number," +

"c.lesson\_start," +

"c.lesson\_week," +

"c.start\_time," +

"ifnull(d.id,'') as homeworkid,"+

"d.lessionDetailId,"+

"d.homeworkName,"+

"d.homeworkInfo,"+

"d.startTime,"+

"d.endTime,"+

"d.createTime,"+

"d.homeworkDesc,"+

"ifnull(e.`name`,'') as lname," +

"f.teacher\_name " +

"from " +

"common\_stuclass as a," +

"common\_student as b," +

"common\_lesson\_detail as c," +

"front\_course\_homework as d," +

"common\_lesson as e," +

"common\_teacher as f " +

"where " +

"a.id = b.class\_id and " +

"c.stuclass\_id = a.id and " +

"d.lessionDetailId = c.id and "+

"e.id = c.lesson\_id and " +

"e.teacher\_id = f.id and " +

"b.id = ? and " +

"c.start\_time > ? and "+

"c.end\_time < ? ";

Query query = getSession().createSQLQuery(queryString);

query.setString(0, classid);

query.setString(1, startdate);

query.setString(2, enddate);

**return** query.setResultTransformer(Transformers.*ALIAS\_TO\_ENTITY\_MAP*)

.list();

}

### 2，左查询

**public** List findLessonHomeWorkDetail(String startdate, String enddate,

String userid) {

String sqlString = "SELECT fch.\*,m.lesson\_desc,m.lesson\_id from front\_course\_homework as fch , (SELECT cld.\* from common\_lesson\_detail cld left JOIN common\_stu\_lesson as csl on cld.id = csl.lesson\_detail\_id WHERE csl.student\_id = ?) as m WHERE fch.lessionDetailId = m.id and fch.startTime > ? and fch.startTime < ?";

Query query = getSession().createSQLQuery(sqlString);

query.setString(0, userid);

query.setString(1, startdate);

query.setString(2, enddate);

**return** query.setResultTransformer(Transformers.*ALIAS\_TO\_ENTITY\_MAP*)

.list();

}

**public** String findTeacherName(String lessonid) {

String sqlString = "SELECT ct.teacher\_name from common\_teacher ct LEFT JOIN common\_lesson cl on ct.id = cl.teacher\_id WHERE cl.id = '"

+ lessonid + "'";

Query query = getSession().createSQLQuery(sqlString);

List list = query

.setResultTransformer(Transformers.*ALIAS\_TO\_ENTITY\_MAP*).list();

String result = **null**;

**try** {

result = ((HashMap<String, String>) list.get(0))

.get("teacher\_name");

} **catch** (Exception e) {

// **TODO**: handle exception

}

**return** result;

}

### 3，只查询条数

SELECT COUNT(\*) FROM common\_resource\_collections WHERE frontuserId = '40288228454f4d1c01454f4d805a0001'

### 4，查询前10条数据

SELECT \* FROM common\_resource\_collections WHERE frontuserId = '40288228454f4d1c01454f4d805a0001' limit 0,10

## 查看应用签名信息

jarsigner -verify -verbose -certs xx.apk

## 查看签名md5值

Keytool –list –v androiddebugkey –keystore debug.keystore

keytool -list -v -alias androiddebugkey -keystore C:\Users\ZLQ\.android\debug.keystore

androiddebugkey 是签名别名。

## ScrollView使用小技巧

### 1，移动到最底端

scrollView.fullScroll(ScrollView.FOCUS\_DOWN);

### 2，填充屏幕

android:fillViewport="true"

## Listview设置背景

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<selector xmlns:android=*"http://schemas.android.com/apk/res/android"*>

<item android:drawable=*"@color/green"* android:state\_pressed=*"true"* android:state\_selected=*"true"*/>

<item android:drawable=*"@android:color/darker\_gray"* android:state\_pressed=*"false"* android:state\_selected=*"true"*/>

<item android:drawable=*"@android:color/white"* android:state\_pressed=*"false"* android:state\_selected=*"false"*/>

</selector>

## 代码中设置android:layout\_centerInParent属性

layoutParams.addRule(RelativeLayout.CENTER\_IN\_PARENT);