

**android:autoLink** 设置是否当文本为URL链接/email/电话号码/map时，文本显示为可点击的链接。可选值(none/web/email/phone/map/all)

**android:autoText** 如果设置，将自动执行输入值的拼写纠正。此处无效果，在显示输入法并输入的时候起作用。

**android:bufferType** 指定getText()方式取得的文本类别。选项editable 类似于StringBuilder可追加字符， 也就是说getText后可调用append方法设置文本内容。**spannable** 则可在给定的字符区域使用样式，参见[这里1](#)、[这里2](#)。

**android:capitalize** 设置英文字母大写类型。此处无效果，需要弹出输入法才能看得到，参见EditView此属性说明。

**android:cursorVisible** 设定光标为显示/隐藏，默认显示。

**android:digits** 设置允许输入哪些字符。如“1234567890.+\*/%\\n()”

**android:drawableBottom** 在text的下方输出一个drawable，如图片。如果指定一个颜色的话会把text的背景设为该颜色，并且同时和background使用时覆盖后者。

**android:drawableLeft** 在text的左边输出一个drawable，如图片。

**android:drawablePadding** 设置text与drawable(图片)的间隔，与drawableLeft、drawableRight、drawableTop、drawableBottom一起使用，可设置为负数，单独使用没有效果。

**android:drawableRight** 在text的右边输出一个drawable，如图片。 **android:drawableTop** 在text的正上方输出一个drawable，如图片。

**android:editable** 设置是否可编辑。这里无效果，参见EditView。

**android:editorExtras** 设置文本的额外的输入数据。在EditView再讨论。

**android:ellipsize** 设置当文字过长时,该控件该如何显示。有如下值设置：“start”——省略号显示在开头；“end”——省略号显示在结尾；“middle”——省略号显示在中间；“marquee”——以跑马灯的方式显示(动画横向移动) **android:freezesText** 设置保存文本的内容以及光标的位置。参见：[这里](#)。

**android:gravity** 设置文本位置，如设置成“center”，文本将居中显示。

**android:hint** Text为空时显示的文字提示信息，可通过textColorHint设置提示信息的颜色。比较奇怪的是TextView本来就相当于Label，怎么会不设置Text?！

**android:imeOptions** 附加功能，设置右下角IME动作与编辑框相关的动作，如actionDone右下角将显示一个“完成”，而不设置默认是一个回车符号。这个在EditView中再详细说明，此处无用。

**android:imeActionId** 设置IME动作ID。在EditView再做说明，可以先看这篇帖子：[这里](#)。 **android:imeActionLabel** 设置IME动作标签。在EditView再做说明。

**android:includeFontPadding** 设置文本是否包含顶部和底部额外空白，默认为true。 **android:inputMethod** 为文本指定输入法，需要完全限定名（完整的包名）。例如：`com.google.android.inputmethod.pinyin`，但是这里报错找不到。

**android:inputType** 设置文本的类型，用于帮助输入法显示合适的键盘类型。在EditView中再详细说明，这里无效果。 **android:linksClickable** 设置链接是否点击连接，即使设置了autoLink。

**android:inputType**未设置文本的类型，让输入法选择合适的软键盘的

**android:marqueeRepeatLimit** 在ellipsize指定marquee的情况下，设置重复滚动的次数，当设置为 `marquee_forever`时表示无限次。

**android:ems** 设置TextView的宽度为N个字符的宽度。这里测试为一个汉字字符宽度，如图：

**android:maxEms** 设置TextView的宽度为最长为N个字符的宽度。与ems同时使用时覆盖ems选项。

**android:minEms** 设置TextView的宽度为最短为N个字符的宽度。与ems同时使用时覆盖ems选项。 **android:maxLength** 限制显示的文本长度，超出部分不显示。

**android:lines** 设置文本的行数，设置两行就显示两行，即使第二行没有数据。

**android:maxLines** 设置文本的最大显示行数，与width或者layout\_width结合使用，超出部分自动换行，超出行数将不显示。 **android:minLines** 设置文本的最小行数，与lines类似。

**android:lineSpacingExtra** 设置行间距。 **android:lineSpacingMultiplier** 设置行间距的倍数。如“1.2”

**android:numeric** 如果被设置，该TextView有一个数字输入法。此处无用，设置后唯一效果是TextView有点击效果，此属性在EditView将详细说明。

**android:password** 以小点“.”显示文本

**android:phoneNumber** 设置为电话号码的输入方式。

**android:privateImeOptions** 设置输入法选项，此处无用，在EditText将进一步讨论。

**android:scrollHorizontally** 设置文本超出TextView的宽度的情况下，是否出现横拉条。

**android:selectAllOnFocus** 如果文本是可选择的，让他获取焦点而不是将光标移动到文本的开始位置或者末尾位置。TextView中设置后无效果。

**android:shadowColor** 指定文本阴影的颜色，需要与shadowRadius一起使用。效果： **android:shadowDx** 设置阴影横向坐标开始位置。

**android:shadowDy** 设置阴影纵向坐标开始位置。

**android:shadowRadius** 设置阴影的半径。设置为0.1就变成字体的颜色了，一般设置为3.0的效果比较好。

**android:singleLine** 设置单行显示。如果和layout\_width一起使用，当文本不能全部显示时，后面用“...”来表示。如**android:text="test\_ singleLine "** **android:singleLine="true"** **android:layout\_width="20dp"**将只显示“t...”。如果不设置singleLine或者设置为false，文本将自动换行 **android:text** 设置显示文本。 **android:textAppearance** 设置文字外观。如“?**android:attr/textAppearanceLargeInverse** ”这里引用的是系统自带的一个外观，? 表示系统是否有这种外观，否则使用默认的外观。可设置的值如下：`textAppearanceButton/textAppearanceInverse/textAppearanceLarge/textAppearanceLargeInverse/textAppearanceMedium/textAppearanceMediumInverse/textAppearanceSmall/textAppearanceSmallInverse` **android:textColor** 设置文本颜色

**android:textColorHighlight** 被选中文字的底色，默认为蓝色 **android:textColorHint** 设置提示信息文字的颜色，默认为灰色。与hint一起使用。

`android:textColorLink` 文字链接的颜色. `android:textScaleX` 设置文字之间间隔, 默认为1.0f. 分别设置0.5f/1.0f/1.5f/2.0f效果如下:

`android:textSize` 设置文字大小, 推荐度量单位"sp", 如"15sp" `android:textStyle` 设置字形[bold(粗体) 0, italic(斜体) 1, bolditalic(又粗又斜) 2] 可以设置一个或多个, 用"|"隔开  
`android:typeface` 设置文本字体, 必须是以下常量值之一: normal 0, sans 1, serif 2, monospace(等宽字体) 3] `android:height` 设置文本区域的高度, 支持度量单位: px(像素)/dp/sp/in/mm(毫米)

`android:maxHeight` 设置文本区域的最大高度

`android:minHeight` 设置文本区域的最小高度

`android:width` 设置文本区域的宽度, 支持度量单位: px(像素)/dp/sp/in/mm(毫米), 与layout\_width的区别看这里。

`android:maxLength` 设置文本区域的最大长度 `android:minWidth` 设置文本区域的最小宽度

248 434

44 231

186-74=112

112/2 \* 2/3 = 37

186-113 = 73

73/3 = 24

100 174

304 417

EditText继承关系: View-->TextView-->EditText

EditText的属性很多, 这里介绍几个:

android:hint="请输入数字!" //设置显示在空间上的提示信息

android:numeric="integer" //设置只能输入整数, 如果是小数则是: decimal

android:singleLine="true" //设置单行输入, 一旦设置为true, 则文字不会自动换行。

android:password="true" //设置只能输入密码

android:textColor = "#ff8c00" //字体颜色

android:textStyle="bold" //字体, bold, italic, bolditalic

android:textSize="20dp" //大小

android:capitalize = "characters" //以大写字母写

android:textAlign="center" //EditText没有这个属性. 但TextView有, 居中

android:textColorHighlight="#c0c0c0" //被选中文字的底色, 默认为蓝色

android:textColorHint="#ff0000" //设置提示信息文字的颜色, 默认为灰色android:textScaleX="1.5" //控制字与字之间的间距

android:typeface="monospace" //字型, normal, sans, serif, monospace

android:background="@null" //背景, 这里没有, 指透明

android:layout\_weight="1" //权重, 控制控件之间的地位, 在控制控件显示的大小时蛮有用的。

android:textAppearance="@android:attr/textAppearanceLargeInverse" //文字外观

android:layout\_gravity="center\_vertical" //设置控件显示的位置: 默认top, 这里居中显示, 还有bottom

android:gray="top" //多行中指针对第一行第一位置

et.setSelection(et.length()); //调整光标到最后一行

android:autoText //自动拼写帮助

android:capitalize //首字母大写

android:digits //设置只接受某些数字

Android: singleLine//是否单行或者多行, 回车是离开文本框还是文本框增加新行

android: numeric //只接受数字

android: phoneNumber //输入电话号码

android: editable //是否可编辑

android:autoLink="all" //设置文本超链接样式当点击网址时, 跳向该网址

android:textAppearance="@android:attr/textAppearanceLargeInverse" //文字外观, 这里引用的是系统自带的一个外观, ? 表示系统是否有这种外观, 否则使用默认的外观。不知道这样理解对不对?

属性名称描述

android:autoLink设置是否当文本为URL链接/email/电话号码/map时, 文本显示为可点击的链接。可选值(none/web/email/phone/map/all)

android:autoText如果设置, 将自动执行输入值的拼写纠正。此处无效果, 在显示输入法并输入的时候起作用。

android:bufferType指定getText()方式取得的文本类别。选项editable 类似于StringBuilder可追加字符,

也就是说getText后可调用append方法设置文本内容。spannable 则可在给定的字符区域使用样式, 参见这里1、这里2。

android:capitalize设置英文字母大写类型。此处无效果, 需要弹出输入法才能看得到, 参见EditText此属性说明。

android:cursorVisible设定光标为显示/隐藏, 默认显示。

android:digits设置允许输入哪些字符。如"1234567890.-\*/% ()"

android:drawableBottom在text的下方输出一个drawable, 如图片。如果指定一个颜色的话会把text的背景设为该颜色, 并且同时和background使用时覆盖后者。

android:drawableLeft在text的左边输出一个drawable, 如图片。

android:drawablePadding设置text与drawable(图片)的间隔, 与drawableLeft、drawableRight、drawableTop、drawableBottom一起使用, 可设置为负数, 单独使用没有效果。

android:drawableRight在text的右边输出一个drawable, 如图片。

android:drawableTop在text的正上方输出一个drawable, 如图片。

android:editable设置是否可编辑。这里无效果, 参见EditText。

android:editorExtras设置文本的额外的输入数据。在EditText再讨论。

android:ellipsize设置当文字过长时,该控件该如何显示。有如下值设置: "start"—?省略号显示在开头;"end"—?省略号显示在结尾;"middle"—?省略号显示在中间;"marquee"—?以跑马灯的方式显示(动画横向往右移动)

android:freezesText设置保存文本的内容以及光标的位置。参见: 这里。

android:gravity设置文本位置, 如设置成"center", 文本将居中显示。

android:hintText为时空时显示的文字提示信息, 可通过textColorHint设置提示信息的颜色。此属性在EditText中使用, 但是这里也可以用。

android:imeOptions附加功能, 设置右下角IME动作与编辑框相关的动作, 如actionDone右下角将显示一个"完成", 而不设置默认是一个回车符号。这个在EditText中再详细说明, 此处无用。

android:imeActionId设置IME动作ID。在EditText再做说明, 可以先看这篇帖子: 这里。

android:imeActionLabel设置IME动作标签。在EditText再做说明。

android:includeFontPadding设置文本是否包含顶部和底部额外空白, 默认为true。

android:inputMethod为文本指定输入法, 需要完全限定名(完整的包名)。例如: com.google.android.inputmethod.pinyin, 但是这里报错找不到。

android:inputType设置文本的类型，用于帮助输入法显示合适的键盘类型。在EditText中再详细说明，这里无效果。

android:linksClickable设置链接是否点击连接，即使设置了autoLink。

android:marqueeRepeatLimit在ellipsize指定marquee的情况下，设置重复滚动的次数。当设置为marquee\_forever时表示无限次。

android:ems设置TextView的宽度为N个字符的宽度。这里测试为一个汉字字符宽度，如图：

android:maxEms设置TextView的宽度为最长为N个字符的宽度。与ems同时使用时覆盖ems选项。

android:minEms设置TextView的宽度为最短为N个字符的宽度。与ems同时使用时覆盖ems选项。

android:maxLength限制显示的文本长度，超出部分不显示。

android:lines设置文本的行数，设置两行就显示两行，即使第二行没有数据。

android:maxLines设置文本的最大显示行数，与width或者layout\_width结合使用，超出部分自动换行，超出行数将不显示。

android:minLines设置文本的最小行数，与lines类似。

android:lineSpacingExtra设置行间距。

android:lineSpacingMultiplier设置行间距的倍数，如“1.2”

android:numeric如果被设置，该TextView有一个数字输入法。此处无用，设置后唯一效果是TextView有点击效果，此属性在EdtiView将详细说明。

android:password以小点“.”显示文本

android:phoneNumber设置为电话号码的输入方式。

android:privateImeOptions设置输入法选项，此处无用，在EditText将进一步讨论。

android:scrollHorizontally设置文本超出TextView的宽度的情况下，是否出现横拉条。

android:selectAllOnFocus如果文本是可选择的，让他获取焦点而不是将光标移动为文本的开始位置或者末尾位置。TextView中设置后无效果。

android:shadowColor指定文本阴影的颜色，需要与shadowRadius一起使用。效果：

android:shadowDx设置阴影横向坐标开始位置。

android:shadowDy设置阴影纵向坐标开始位置。

android:shadowRadius设置阴影的半径。设置为0.1就变成字体的颜色了，一般设置为3.0的效果比较好。

android:singleLine设置单行显示。如果和layout\_width一起使用，当文本不能全部显示时，后面用“...”来表示。如android:text="test\_singleLine " android:singleLine="true" android:layout\_width="20dp"将只显示“t...”。如果不设置singleLine或者设置为false，文本将自动换行

android:shadowDx设置阴影横向坐标开始位置。

android:shadowDy设置阴影纵向坐标开始位置。

android:shadowRadius设置阴影的半径。设置为0.1就变成字体的颜色了，一般设置为3.0的效果比较好。

android:singleLine设置单行显示。如果和layout\_width一起使用，当文本不能全部显示时，后面用“...”来表示。如android:text="test\_singleLine " android:singleLine="true" android:layout\_width="20dp"将只显示“t...”。如果不设置singleLine或者设置为false，文本将自动换行

android:text设置显示文本。

android:textSize设置文字大小，推荐度量单位“sp”，如“15sp”

android:textStyle设置字形[bold(粗体) 0, italic(斜体) 1, bolditalic(又粗又斜) 2] 可以设置一个或多个，用“|”隔开

android:typeface设置文本字体，必须是以下常量值之一：normal 0, sans 1, serif 2, monospace(等宽字体) 3]

android:height设置文本区域的高度，支持度量单位：px(像素)/dp/sp/in/mm(毫米)

android:maxHeight设置文本区域的最大高度

android:minHeight设置文本区域的最小高度

android:width设置文本区域的宽度，支持度量单位：px(像素)/dp/sp/in/mm(毫米)，与layout\_width的区别看这里。

android:maxWidth设置文本区域的最大宽度

android:minWidth设置文本区域的最小宽度

android:textAppearance设置文字外观。如“?android:attr/textAppearanceLargeInverse

“这里引用的是系统自带的一个外观，?表示系统是否有这种外观，否则使用默认的外观。可设置的值如下：textAppearanceButton/textAppearanceInverse/textAppearanceLarge/textAppearanceLargeInverse/textAppearanceMedium/textAppearanceMediumInverse/textAppearanceSmall/textAppearanceSmallInverse

android:textAppearance设置文字外观。如“?android:attr/textAppearanceLargeInverse

“这里引用的是系统自带的一个外观，?表示系统是否有这种外观，否则使用默认的外观。可设置的值如下：textAppearanceButton/textAppearanceInverse/textAppearanceLarge/textAppearanceLargeInverse/textAppearanceMedium/textAppearanceMediumInverse

这里用最甜蜜的幸福,最博大的付出,最深刻的背叛,最强烈的绝望,最勇敢的坚强,最有力的反击!站在是非之外，看红尘最美不过月华清明、百花围簇!备注：在这个道德沦丧的时代，如果你想背弃誓言，轻易转身，那就不用再回头，因为没有人站在原地等你!

当旋转手机时，如果想要保存EditText的文本的内容，有三种方法：

一：

只要这个EditText设置了android: id属性就会自动保存。

二：

可以设置EditText的android:freezesText="true"，再到相应的Activity中加上如下两个方法：

```
CharSequence getSavedText(){
    return ((TextView)findViewById(R.id.saved)).getText();
}

void setSavedText(CharSequence text){
    ((TextView)findViewById(R.id.saved)).setText(text);
}
```

如果想在退出这个Activity时，也能保存EditText的内容：

这时可以用state机制来保存。

```
mSearchInput.setImeOptions(EditorInfo.IME_ACTION_DONE);
mSearchInput.setImeActionLabel(getString(R.string.search_action_label), EditorInfo.IME_ACTION_DONE);
mSearchInput.setImeOptions(EditorInfo.IME_ACTION_SEARCH);
```

```
mSearchInput.setOnEditorActionListener(new TextView.OnEditorActionListener() {
    @Override
    public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {
        if (actionId == EditorInfo.IME_ACTION_DONE) {
            performSearch();
            return true;
        }
        return false;
    }
});
```

Romain Guy write a little info about a **ScrollView** attribute that is missing from documentation : **android:fillViewport="true"** .

It must be set to **ScrollView** and has the following effect : when set to true, this attribute causes the scroll view's child to expand to the height of the **ScrollView** if needed. When the child is taller than the **ScrollView**, the attribute has no effect.

当你想让一个高度值不足scrollview的子控件fillparent的时候，单独的定义 android:layout\_height="fill\_parent"是不起作用的，必须加上fillviewport属性，当子控件的高度值大于 scrollview的高度时，这个标签就没有任何意义了。

圆角textView：

首先在Drawable文件下建一个corner\_view.xml文件：

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
```

solid为填充颜色，如果不写，将是默认的黑色，我这里设置成白色

```
<solid android:color="#ffffff" />
<corners android:radius="10dp" />
<padding android:left="5dp" android:top="5dp"
android:right="5dp" android:bottom="5dp" />
</shape>
```

然后把你的textView实例对象调用mSignature.setBackgroundResource(R.drawable.corner\_view);即可。

```
<?xml version="1.0" encoding="UTF-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#FFFFFF" />
    <corners android:topLeftRadius="10px"
        android:topRightRadius="10px" android:bottomLeftRadius="1px"
        android:bottomRightRadius="1px"/>
    <padding android:left="0dp" android:top="0dp" android:right="0dp"
        android:bottom="0dp" />
</shape>
```

```
gv.setSelector(new ColorDrawable(Color.TRANSPARENT));
```

1、 ContextMenu:

- 2、 菜单
- 3、 progressbar:
- 5、 adb命令 / 进入sqlite
- 6、 git命令
- 13、 网络
- 14、 搜索

1、 ContextMenu:

[一]:

```
@Override
    public void onCreateContextMenu(ContextMenu menu, View v,
        ContextMenuInfo menuInfo) {
        super.onCreateContextMenu(menu, v, menuInfo);

        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.context_menu, menu);
    }
[二]:
```

```
@Override
    public boolean onContextItemSelected(MenuItem item) {
        AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();

        switch (item.getItemId()) {
            case R.id.item_logout:
                // break;
            case R.id.item_update:
                // Toast.makeText(this, "update", 1).show();
                break;
        }
        return super.onContextItemSelected(item);
    }
[三]: 在menu目录下建立:
```

```
<?xml version="1.0" encoding="utf-8"?>
<menu
    xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/item_logout" android:title="注销"></item>
    <item android:id="@+id/item_update" android:title="更新"></item>
</menu>
```

2、 菜单

```
@Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main_menu, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.item_clear_memory_cache:
                imageLoader.clearMemoryCache();
                break;
            case R.id.item_clear_disc_cache:
                imageLoader.clearDiscCache();
                break;
            case R.id.item_exit:
                dialog = new MyDialog(this, null).title("警告").message("确定退出吗? ")
                    .positiveButton("OK", null).negativeButton("Cancle", null)
                    .setListener(myDialogClickListener);
                dialog.show();
                break;
            default:
                return false;
        }
        return true;
    }
}
```

在menu目录下:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/item_exit"
        android:title="退出"
    />
    <item
        android:id="@+id/item_clear_memory_cache"
        android:title="清除内存缓存"/>

    <item
        android:id="@+id/item_clear_disc_cache"
        android:title="清除磁盘缓存"
    />
</menu>
```

3、 progressbar:

```

<ProgressBar
    android:id="@+id/progressBar1"
    style="@android:style/Widget.ProgressBar.Inverse"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="400dp"
    android:indeterminate="false"
    android:indeterminateDrawable="@drawable/progressbar_color" />

```

对应的图片:

```

<?xml version="1.0" encoding="utf-8"?>
<rotate xmlns:android="http://schemas.android.com/apk/res/android"
    android:fromDegrees="0"
    android:pivotX="50%"
    android:pivotY="50%"
    android:toDegrees="360" >

```

```

<shape
    android:innerRadiusRatio="3"
    android:shape="ring"
    android:thicknessRatio="8"
    android:useLevel="false"
    >
    <gradient
        android:centerColor="#7B68EE"
        android:centerY="0.50"
        android:endColor="#912CEE"
        android:startColor="#9F79EE"
        android:type="sweep"
        android:useLevel="false"
    />
</shape>
</rotate>

```

#### 4. 布局: 平均分配

```

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/like_category_bar"
    android:gravity="center_vertical"
    android:orientation="horizontal" >

    <ImageButton
        android:id="@+id/item_detail_webview_back"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/webview_button_close"
        android:layout_marginLeft="10dp"/>

    <View
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="fill_parent"
    />

    <ImageButton
        android:id="@+id/item_detail_webview_backward"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/webview_button_arrowleft_disable" />

<View
    android:layout_width="0dp"
    android:layout_weight="1"
    android:layout_height="fill_parent"
/>

    <ImageButton
        android:id="@+id/item_detail_webview_forward"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/webview_button_arrowright_disable" />

<View
    android:layout_width="0dp"
    android:layout_weight="1"
    android:layout_height="fill_parent"
/>

    <ImageButton
        android:id="@+id/item_detail_webview_refresh"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/webview_refresh"
        android:layout_marginRight="10dp" />

</LinearLayout>

```

#### 5. 进入sqlite

```

cd /Users/wangliang/Downloads/android-sdk-macosx/platform-tools
./adb -s emulator-5554 shell
adb shell
sqlite3 data/data/com.guoku/databases/mydata.db
sqlite3 data/data/com.guoku/databases/Cards
sqlite3 data/data/wangliang.test/databases/my.db
.tables
.scheme 表名
.schema

```

联系人:

```

./adb -s emulator-5554 shell
sqlite3 data/data/com.android.providers.contacts/databases/contacts2.db
sqlite> .tables
_sync_state          settings
_sync_state_metadata status_updates
accounts             v1_settings
activities           view_contacts
agg_exceptions       view_contacts_restricted
android_metadata     view_data
calls                view_data_restricted
contact_entities_view view_groups
contact_entities_view_restricted view_raw_contacts
contacts             view_raw_contacts_restricted
data                 view_v1_contact_methods
groups               view_v1_extensions
mimetypes            view_v1_group_membership
name_lookup          view_v1_groups
nickname_lookup      view_v1_organizations
packages             view_v1_people
phone_lookup         view_v1_phones

```

```
properties          view_v1_photos
row_contacts
```

```
sqlite> .schema contacts
```

```
CREATE TABLE contacts (
    _id INTEGER PRIMARY KEY AUTOINCREMENT,
    name_raw_contact_id INTEGER REFERENCES raw_contacts(_id),
    photo_id INTEGER REFERENCES data(_id),
    custom_ringtone TEXT,
    send_to_voicemail INTEGER NOT NULL DEFAULT 0,
    times_contacted INTEGER NOT NULL DEFAULT 0,
    last_time_contacted INTEGER,
    starred INTEGER NOT NULL DEFAULT 0,
    in_visible_group INTEGER NOT NULL DEFAULT 1,
    has_phone_number INTEGER NOT NULL DEFAULT 0,
    lookup TEXT,
    status_update_id INTEGER REFERENCES data(_id),
    single_is_restricted INTEGER NOT NULL DEFAULT 0
);
CREATE INDEX contacts_has_phone_index ON contacts (has_phone_number);
CREATE INDEX contacts_name_raw_contact_id_index ON contacts (name_raw_contact_id);
CREATE INDEX contacts_restricted_index ON contacts (single_is_restricted);
CREATE INDEX contacts_visible_index ON contacts (in_visible_group);
```

## 1.contacts 表

\_id : 表的ID, 主要用于其它表通过contacts 表中的ID可以查到相应的数据。

display\_name: 联系人名称

photo\_id:头像的ID, 如果没有设置联系人头像, 这个字段就为空

times\_contacted:通话记录的次数

last\_time\_contacted: 最后的通话时间

lookup :是一个持久化的储存 因为用户可能会改名字 但是它改不了lookup

```
sqlite> .schema data
```

```
CREATE TABLE data (
    _id INTEGER PRIMARY KEY AUTOINCREMENT,
    package_id INTEGER REFERENCES package(_id),
    mimetype_id INTEGER REFERENCES mimetype(_id) NOT NULL,
    raw_contact_id INTEGER REFERENCES raw_contacts(_id) NOT NULL,
    is_primary INTEGER NOT NULL DEFAULT 0,
    is_super_primary INTEGER NOT NULL DEFAULT 0,
    data_version INTEGER NOT NULL DEFAULT 0,
    data1 TEXT,data2 TEXT,data3 TEXT,data4 TEXT,data5 TEXT,data6 TEXT,data7 TEXT,data8 TEXT,data9 TEXT,data10 TEXT,data11 TEXT,data12 TEXT,data13 TEXT,data14 TEXT,data15
    TEXT,
    data_sync1 TEXT, data_sync2 TEXT, data_sync3 TEXT, data_sync4 TEXT
);
CREATE INDEX data_mimetype_data1_index ON data (mimetype_id,data1);
CREATE INDEX data_raw_contact_id ON data (raw_contact_id);
CREATE TRIGGER data_deleted BEFORE DELETE ON data
BEGIN
    UPDATE raw_contacts SET version=version+1 WHERE _id=OLD.raw_contact_id;
    DELETE FROM phone_lookup WHERE data_id=OLD._id;
    DELETE FROM status_updates WHERE status_update.data_id=OLD._id;
    DELETE FROM name_lookup WHERE data_id=OLD._id;
END;
CREATE TRIGGER data_updated AFTER UPDATE ON data
BEGIN
    UPDATE data SET data_version=OLD.data_version+1 WHERE _id=OLD._id;
    UPDATE raw_contacts SET version=version+1 WHERE _id=OLD.raw_contact_id;
END;
```

## 2.data表

raw\_contact\_id: 通过raw\_contact\_id可以找到 raw\_contact表中相对的数据。

data1 到 data15 这里保存着联系人的信息 联系人名称 联系人电话号码 电子邮件 备注 等等。

```
sqlite> .schema phone_lookup
```

```
CREATE TABLE phone_lookup (
    data_id INTEGER PRIMARY KEY REFERENCES data(_id) NOT NULL,
    raw_contact_id INTEGER REFERENCES raw_contacts(_id) NOT NULL,
    normalized_number TEXT NOT NULL,
    min_match TEXT NOT NULL
);
CREATE INDEX phone_lookup_index ON phone_lookup (normalized_number,raw_contact_id,data_id);
CREATE INDEX phone_lookup_min_match_index ON phone_lookup (min_match,raw_contact_id,data_id);
```

## 3.phone\_lookup\_表

data\_id : 通过data\_id可以找到 data表中相对的数据。

raw\_contact\_id : 通过raw\_contact\_id 可以找到 raw\_contact\_表中相对的数据。

normalized\_number: 这个字段就比较有意思了, 它是将每个电话号码逆序排列。

```
sqlite> .schema raw_contacts
```

```
CREATE TABLE raw_contacts (
    _id INTEGER PRIMARY KEY AUTOINCREMENT,
    is_restricted INTEGER DEFAULT 0,
    account_name STRING DEFAULT NULL,
    account_type STRING DEFAULT NULL,
    source_id TEXT,
    version INTEGER NOT NULL DEFAULT 1,
    dirty INTEGER NOT NULL DEFAULT 0,
    deleted INTEGER NOT NULL DEFAULT 0,
    contact_id INTEGER REFERENCES contacts(_id),
    aggregation_mode INTEGER NOT NULL DEFAULT 0,
    aggregation_needed INTEGER NOT NULL DEFAULT 1,
    custom_ringtone TEXT,send_to_voicemail INTEGER NOT NULL DEFAULT 0,
    times_contacted INTEGER NOT NULL DEFAULT 0,
    last_time_contacted INTEGER,
    starred INTEGER NOT NULL DEFAULT 0,
    display_name TEXT,
    display_name_alt TEXT,
    display_name_source INTEGER NOT NULL DEFAULT 0,
    phonetic_name TEXT,
    phonetic_name_style TEXT,
    sort_key TEXT COLLATE PHONETIC,
    sort_key_alt TEXT COLLATE PHONETIC,
    name_verified INTEGER NOT NULL DEFAULT 0,
    contact_in_visible_group INTEGER NOT NULL DEFAULT 0,
    sync1 TEXT, sync2 TEXT, sync3 TEXT, sync4 TEXT );
```

```

CREATE INDEX raw_contact_sort_key1_index ON raw_contacts (contact_in_visible_group,sort_key);
CREATE INDEX raw_contact_sort_key2_index ON raw_contacts (contact_in_visible_group,sort_key_alt);
CREATE INDEX raw_contacts_contact_id_index ON raw_contacts (contact_id);
CREATE INDEX raw_contacts_source_id_index ON raw_contacts (sourceid, account_type, account_name);
CREATE TRIGGER raw_contacts_deleted BEFORE DELETE ON raw_contacts
BEGIN
    DELETE FROM data WHERE raw_contact_id=OLD._id;
    DELETE FROM agg_exceptions WHERE raw_contact_id1=OLD._id OR raw_contact_id2=OLD._id;
    DELETE FROM contacts WHERE _id=OLD.contact_id AND (SELECT COUNT(*) FROM raw_contacts WHERE contact_id=OLD.contact_id)=1;
END;
CREATE TRIGGER raw_contacts_marked_deleted AFTER UPDATE ON raw_contacts
BEGIN
    UPDATE raw_contacts SET version=OLD.version+1 WHERE _id=OLD._id AND NEW.deleted!= OLD.deleted;
END;

```

#### 4.raw\_contact表

```

raw_contacts表在raw_contacts表的基础上增加以下列:
contact_in_visible_group, sort_key, sort_key_alt,
sourceid, account_type, account_name,
deleted, display_name, last_time_contacts

```

activity-alias <http://blog.csdn.net/actelecity/article/details/6561393>

<http://my.oschina.net/zhangqingcai/blog/42085>

联系人: <http://blog.csdn.net/furongkang/article/details/6832082>

<http://www.cnblogs.com/3dant/archive/2010/12/12/1904031.html>

version :版本号, 用于监听变化

deleted :删除标志, 0为默认 1表示这行数据已经删除

display\_name :联系人名称

last\_time\_contacts :最后联系的时间

#### sqlite> schema mimetypes

```

CREATE TABLE mimetypes (
  _id INTEGER PRIMARY KEY AUTOINCREMENT,
  mimetype TEXT NOT NULL
);
CREATE UNIQUE INDEX mime_type ON mimetypes (_mimetype);

```

```

sqlite> select * from mimetypes;
1|vnd.android.cursor.item/email_v2
2|vnd.android.cursor.item/im
3|vnd.android.cursor.item/postal-address_v2
4|vnd.android.cursor.item/photo
5|vnd.android.cursor.item/phone_v2
6|vnd.android.cursor.item/name
7|vnd.android.cursor.item/organization
8|vnd.android.cursor.item/nickname
9|vnd.android.cursor.item/group_membership

```

android 1.6中, SIM的provider是IccProvider。既然是provider, 就和操作Contacts的provider类似的, 只是有一些微小差别。IccProvider的Uri是content://icc/and, 可以通过这个Uri来操作SIM卡。当然, 使用时需要转换为android的Uri对象。android中对SIM卡操作的支持很简单, 具体的查询、插入、更新和删除操作如下面的代码段

#### 一、查询联系人

查询时, 只支持获取获取联系人, 即query()函数的后面几个参数都为null (其它值不起作用)。同时, 也不支持类似content://icc/and/0的查询。

```

public void SimQuery(Activity activity) {
    Uri uri = Uri.parse("content://icc/adn");
    Cursor cursor = activity.getContentResolver().query(uri, null, null, null, null);
    Log.d("1023", ">>>>>>" + cursor.getCount());
    while (cursor.moveToNext()) {
        String id = cursor.getString(cursor.getColumnIndex(People.ID));
        String name = cursor.getString(cursor.getColumnIndex(People.NAME));
        String phoneNumber = cursor.getString(cursor.getColumnIndex(People.NUMBER));
    }
}

```

#### 二、插入联系人

插入联系人只要设置名字和电话号码就可以了, 要注意的是名字对应的是tag, 而不是name。

```

public void SimInsert(Activity activity) {
    Uri uri = Uri.parse("content://icc/adn");
    ContentValues values = new ContentValues();
    values.put("tag", "Jones");
    values.put("number", "10086");
    Uri newSimContactUri = activity.getContentResolver().insert(uri, values);
}

```

#### 三、更新联系人

更新联系人要注意的是, 它是原先的名字和电话号码来匹配要更新的联系人的, 故要指定4个属性。

```

public void SimUpdate(Activity activity) {
    Uri uri = Uri.parse("content://icc/adn");
    ContentValues values = new ContentValues();
    values.put("tag", "Jones");
    values.put("number", "10086");
    values.put("newTag", "Fred");
    values.put("newNumber", "10000");
    activity.getContentResolver().update(uri, values, null, null);
}

```

#### 四、删除联系人

删除联系人, 同样是以名字和电话号码来匹配的, 故需要在delete的where参数中指定, 下面的例子演示了删除所有SIM卡内的联系人。

```

public void SimDelete(Activity activity) {
    Uri uri = Uri.parse("content://icc/adn");
    Cursor cursor = activity.getContentResolver().query(uri, null, null, null, null);
    while (cursor.moveToNext()) {
        String name = cursor.getString(cursor.getColumnIndex(People.NAME));
        String phoneNumber = cursor.getString(cursor.getColumnIndex(People.NUMBER));
        String where = "tag='" + name + "'";
        where += " AND number='" + phoneNumber + "'";
        activity.getContentResolver().delete(uri, where, null);
    }
}

```

上面的代码我都测过, 可以放心使用。要使上面的例子成功, 则需要添加到SIM卡上的联系人有不同的名字或电话号码, 若有相同的名字和电话号, 可能会带来问题, 这一点我未测过。还有, 要注意的是SIM卡和U盘一样容易坏, 若读写次数较多, 有可能会坏掉。

一般数据采用的固定的静态数据类型，而SQLite采用的是动态数据类型，会根据存入值自动判断，SQLite具有以下五种数据类型：

1. NULL：空值。
2. INTEGER：带符号的整型，具体取决于存入数字的范围大小。
3. REAL：浮点数字，存储为8-byte IEEE浮点数。
4. TEXT：字符串文本。
5. BLOB：二进制对象。

但实际上，sqlite3也接受如下的数据类型：

smallint 16 位元的整数。  
integer 32 位元的整数。  
decimal(p,s) p 精度值和 s 大小的十进制整数，精确值p是指全部有几个数(digits)大小值，s是指小数点後有几位数。如果没有特别指定，则系统会设为 p=5；s=0。  
float 32位元的实数。  
double 64位元的实数。  
char(n) n 长度的字串，n不能超过 254。  
varchar(n) 长度不固定且其最大长度为 n 的字串，n不能超过 4000。  
graphic(n) 和 char(n) 一样，不过其单位是两个字节 double-bytes，n不能超过127。这个形态是为了支援两个字节长度的字体，例如中文字。  
vargraphic(n) 可变长度且其最大长度为 n 的双字节字符串，n不能超过 2000  
date 包含了 年份、月份、日期。  
time 包含了 小时、分钟、秒。  
timestamp 包含了 年、月、日、时、分、秒、千分之一秒。

#### adb shell dumpsys activity

.....

Running activities (most recent first):

模拟器快捷键：./emulator -help-keys

[https://groups.google.com/group/android-developers/browse\\_thread/thread/90355f819a734c3e/29cc72c6702d3c7c?show\\_docid=29cc72c6702d3c7c&pli=1](https://groups.google.com/group/android-developers/browse_thread/thread/90355f819a734c3e/29cc72c6702d3c7c?show_docid=29cc72c6702d3c7c&pli=1)

echo STMPDIR

df -h

TMPDIR=\$HOME/tmp

TMPDIR=\$HOME/tmp SHOME/bin/svserve -r root\_dir -d

6. git命令

git status

git add .

git commit -a -m'commit'

git push

adb install -s(sdcard) -(reinstall)

dab uninstall -k(keep data/di/cache)

oapt 查看apk包、结构、版本

emulator -avd Test -lcpdump capture.pcap生成文件，使用WireShark查看

java (bin目录) 命令

key tool -list -v -keystore keystore文件 查看签名信息

jarsigner -verify apk文件 验证签名

usage: monkey [-p ALLOWED\_PACKAGE [-p ALLOWED\_PACKAGE] ...]

```
[--c MAIN_CATEGORY [-c MAIN_CATEGORY] ...]
[-ignore-crashes] [-ignore-timeouts]
[-ignore-security-exceptions]
[-monitor-native-crashes] [-ignore-native-crashes]
[-kill-process-after-error] [-hprof]
[-pct-touch PERCENT] [-pct-motion PERCENT]
[-pct-backstab PERCENT] [-pct-syskeys PERCENT]
[-pct-nav PERCENT] [-pct-mojomov PERCENT]
[-pct-appswitch PERCENT] [-pct-flip PERCENT]
[-pct-anyevent PERCENT]
[-pkg-blacklist-file PACKAGE_BLACKLIST_FILE]
[-pkg-whitelist-file PACKAGE_WHITELIST_FILE]
[-wait-dbg] [-dbg-no-events]
[-setup scriptfile] [-scriptfile [-scriptfile] ...]
[-port port]
[-s SEED] [-v [-v] ...]
[-throttle MILLISEC] [-randomize-throttle]
[-profile-wait MILLISEC]
[-device-sleep-time MILLISEC]
[-randomize-script]
[-script-log]
[-bugreport]
COUNT
```

dab shell mokey -p 包名  
-p 包名.如果调用了系统的包  
-throttle延迟(毫秒)

adb shell am instrument -w com.testcalculator/android.test.InstrumentationTestRunner

#### 7、适配器

(1) android.widget.SimpleAdapter.SimpleAdapter(Context context, List<? extends Map<String, ?>> data, int resource, String[] from, int[] to)

List<Contact> contacts = getAllContacts();

List<Map<String, Object>> data = new ArrayList<Map<String, Object>>();

for (Contact p : contacts) {

Map<String, Object> map = new HashMap<String, Object>();

//map.put("id", p.getId());没有id字段

map.put("name", p.getName());

map.put("phone", p.getPhone());

map.put("email", p.getEmail());

data.add(map);

}

SimpleAdapter adapter = new SimpleAdapter(//

this, data, R.layout.*listlayout*, // *listlayout*是布局文件

new String[] { "name", "phone", "email" }, //

new int[] { R.id.name, R.id.phone, R.id.email });

(2) android.widget.SimpleCursorAdapter.SimpleCursorAdapter(Context context, int layout, Cursor c, String[] from, int[] to)

PersonService service = new PersonService(this);

Cursor c = service.findPageCursor(1, 40);

SimpleCursorAdapter adapter = new SimpleCursorAdapter(//

this, R.layout.item, c, new String[] { "name", "phone", "balance" }, new int[] { R.id.name, R.id.phone, R.id.balance });

// "name", "phone", "balance"是游标中的字段

listView.setAdapter(adapter);

listView.setOnItemClickListener(new MyCursorListener());

}

private final class MyCursorListener implements OnItemClickListener {

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

// 先从listView中获取adapter，再从adapter中获取Cursor，再将Cursor指向position位置，最后将Cursor返回

Cursor c = (Cursor) listView.getItemAtPosition(position);

Toast.makeText(getApplicationContext(), c.getString(c.getColumnIndex("name")), 0).show();



```

    }
}
其中
public Cursor findPageCursor(int page, int size) {
    SQLiteDatabase db = helper.getReadableDatabase();
    return db.query(/*
        "person", new String[] { "id_id", "name", "phone", "balance" }, null, null, null, null, (page - 1) * size + "," + size);
//这里必须要有_id字段
}

```

例子：最简单的含有复选框的listview

```

public class List11 extends ListActivity {

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

        setListAdapter(new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_multiple_choice, GENRES));

        final ListView listView = getListView();

        listView.setItemsCanFocus(false);
        listView.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
    }

    private static final String[] GENRES = new String[] {
        "Action", "Adventure", "Animation", "Children", "Comedy", "Documentary", "Drama",
        "Foreign", "History", "Independent", "Romance", "Sci-Fi", "Television", "Thriller"
    };
}

```

其中 android.R.layout.simple\_list\_item\_multiple\_choice为:

```

//      <CheckedTextView xmlns:android="http://schemas.android.com/apk/res/android"
//          android:id="@android:id/text1"
//          android:layout_width="match_parent"
//          android:layout_height="?android:attr/listPreferredItemHeight"
//          android:textAppearance="?android:attr/textAppearanceLarge"
//          android:gravity="center_vertical"
//          android:checkMark="?android:attr/listChoiceIndicatorMultiple"
//          android:paddingLeft="6dip"
//          android:paddingRight="6dip"
//      />

```

## 8. dialog

```

@Override
protected Dialog onCreateDialog(int id) {
    switch (id) {
        case DIALOG_SHARE:
            return new AlertDialog.Builder(CardDetail.this)
                .setTitle(R.string.detail_share_dialog_title)
                .setItems(R.array.share_dialog_items,
                    new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog,
                            int which) {
                            Intent intent = null;
                            String[] items = getResources()
                                .getStringArray(
                                    R.array.share_dialog_items);

                            switch (which) {
                                case 0: // 分享到微博

                                    break;

                                default:
                                    break;
                            }
                        }
                    })
                .setNegativeButton("取消",
                    new DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog,
                            int which) {}
                    })
                .create();
    }
    // TODO Auto-generated method stub
    return super.onCreateDialog(id);
}

```

## 9. 动画：图片淡入淡出、翻扑克牌

```

<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator">
    <scale
        android:fromXScale="0.0"
        android:toXScale="1.0"
        android:fromYScale="1.0"
        android:toYScale="1.0"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration="150"/>
</set>

?xml version="1.0" encoding="utf-8"?
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator">
    <alpha
        android:fromAlpha="1.0"
        android:toAlpha="0.2"
        android:duration="500"
    />
</set>

```

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator">
    <rotate
        android:fromDegrees="0"
        android:toDegrees="360"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration="3000"/>
</set>
```

```
imageView.setOnClickListener(new ImgViewListener());

监听器为:
//ImageView点击事件
class ImgViewListener implements OnClickListener {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        //也可通过代码来实现 这个是收缩效果
        AnimationSet animationSet = new AnimationSet(true);
        ScaleAnimation scale = new ScaleAnimation(1, 0.2f, 1, 0.2f,
            Animation.RELATIVE_TO_SELF, 0.5f,
            Animation.RELATIVE_TO_SELF, 0.5f);
        AlphaAnimation alphaAnimation = new AlphaAnimation(1.0f, 0.2f);
        animationSet.addAnimation(scale);
        animationSet.addAnimation(alphaAnimation);
        animationSet.setDuration(400);

        //通过AnimationUtils得到动画配置文件(/res/anim/back_scale.xml)
        Animation animation = AnimationUtils.loadAnimation(Activity01.this, R.anim.alpha_out);
        animationSet.setAnimationListener(new Animation.AnimationListener() {
            @Override
            public void onAnimationStart(Animation animation) {
            }
            @Override
            public void onAnimationRepeat(Animation animation) {
            }
            @Override
            public void onAnimationEnd(Animation animation) {
                if (bool) {
                    imageView.setImageResource(R.drawable.back);
                    bool = false;
                } else {
                    imageView.setImageResource(R.drawable.front);
                    bool = true;
                }
                //通过AnimationUtils得到动画配置文件(/res/anim/front_scale.xml),然后在把动画交给ImageView
                imageView.startAnimation(AnimationUtils.loadAnimation(Activity01.this, R.anim.alpha_in));

                AnimationSet set = new AnimationSet(true);
                ScaleAnimation scale = new ScaleAnimation(0.2f, 1f, 0.2f, 1f,
                    Animation.RELATIVE_TO_SELF, 0.5f,
                    Animation.RELATIVE_TO_SELF, 0.5f);
                AlphaAnimation alphaAnimation = new AlphaAnimation(0.2f, 1f);
                set.addAnimation(scale);
                set.addAnimation(alphaAnimation);
                set.setDuration(1000);
                imageView.startAnimation(set);
            }
        });
        imageView.startAnimation(animationSet);
    }
}
```

#### 10. Toast

```
Toast toast = Toast.makeText(this, null, Toast.LENGTH_LONG);
LinearLayout layout = (LinearLayout) toast.getView();
layout.setOrientation(LinearLayout.HORIZONTAL);

ImageView imageView = new ImageView(this);
imageView.setImageResource(R.drawable.succ);
layout.addView(imageView);

TextView tv = new TextView(this);
tv.setLayoutParams(new LayoutParams(LayoutParams.WRAP_CONTENT, LayoutParams.FILL_PARENT));
tv.setGravity(Gravity.CENTER_VERTICAL);
tv.setPadding(10, 0, 0, 0);
tv.setText(R.string.mood_fine);
layout.addView(tv);
toast.show();
```

#### 11. 包、程序

如何获取Android正在运行的应用和它的pid

```
ActivityManager mActivityManager = (ActivityManager) this.getSystemService(ACTIVITY_SERVICE);
List<ActivityManager.RunningAppProcessInfo> mRunningProcess = mActivityManager.getRunningAppProcesses();
int i = 1;
for (ActivityManager.RunningAppProcessInfo amProcess : mRunningProcess) {
    Log.i("Application", (i++) + "PID: " + amProcess.pid + "(processName=" + amProcess.processName + "UID=" + amProcess.uid + ")");
}
}
```

```
ActivityManager manager = (ActivityManager) getApplicationContext().getSystemService(ACTIVITY_SERVICE);
List<RunningTaskInfo> runningTasks = manager.getRunningTasks(1);
System.out.println("runningTasks.size():" + runningTasks.size());
String packageName = runningTasks.get(0).topActivity.getPackageName();
System.out.println("runningTasks.get(0).numActivities:" + runningTasks.get(0).numActivities);
RunningTaskInfo info = runningTasks.get(0);
```

#### 12. 关于图像

```
// 1.Drawable drawable =
// getResources().getDrawable(R.drawable.ic_launcher);
// 2.BitmapFactory
// 3.BitmapDrawable
// 4.Drawable
// 5.getResources().openRawResource(id)
// 6.getAssets().open(fileName)
```

### 13、自定义View

SurfaceView和View最本质的区别在于，surfaceView是在一个新起的单独线程中可以重新绘制画面而View必须在UI的主线程中更新画面。  
那么在UI的主线程中更新画面 可能会引发问题，比如你更新画面的时间过长，那么你的主UI线程会被你正在画的函数阻塞，那么将无法响应按键，触屏等消息。  
当使用surfaceView 由于是在新的线程中更新画面所以不会阻塞你的UI主线程，但这也带来了另外一个问题，就是事件同步。比如你触屏了一下，你需要surfaceView中 thread处理，一般就需要有一个event queue的设计来保存touch event，这会稍稍复杂一点，因为涉及到线程同步。

使用的SurfaceView的时候，一般情况下来要对其进行创建、销毁，改变时的情况进行监视，这就要用到 SurfaceHolder.Callback。  
class XxxView extends SurfaceView implements SurfaceHolder.Callback {

```
    public void surfaceChanged(SurfaceHolder holder,int format,int width,int height){}
    //顾名思义，在surface的大小发生改变时重绘
    public void surfaceCreated(SurfaceHolder holder){}
    //同上，在创建时激发，一般在这里调用画图的线程。
    public void surfaceDestroyed(SurfaceHolder holder) {}
    //同上，销毁时激发，一般在这里将画图的线程停止、释放。
```

}

SurfaceView的getHolder()函数可以获取SurfaceHolder对象，Surface 就在SurfaceHolder对象内。虽然Surface保存了当前窗口的像素数据，但是在使用过程中是不直接和Surface打交道的，由SurfaceHolder的Canvas lockCanvas()或则Canvas lockCanvas()函数来获取Canvas对象，通过在Canvas上绘制内容来修改Surface中的数据。如果Surface不可编辑或则尚未创建调用该函数会返回null。在 unlockCanvas() 和 lockCanvas()中Surface的内容是不缓存的，所以需要完全重绘Surface的内容。为了提高效率只重绘变化的部分  
则可以调用 lockCanvas(Rect rect)函数来指定一个rect区域，这样该区域外的内容会缓存起来。在调用LockCanvas函数获取Canvas后，SurfaceView会获取Surface的一个同步锁直到调用unlockCanvasAndPost((Canvas canvas)函数才释放该锁，这里的同步机制保证在Surface绘制过程中不会被改变（被摧毁、修改）。

当surfaceCreated()被执行的时候，真正的view才被创建，也就是说之前得到p屏幕的宽和高都为0，是因为初始化会在surfaceCreated ()方法执行以前执行

surfaceview中确实有 onDraw这个方法，但是你surfaceview不会自己去调用

\* surfaceCreated(SurfaceHolder holder): 当Surface第一次创建后会立即调用该函数。程序可以在该函数中做些和绘制界面相关的初始化工作，一般情况下都是在另外的线程来绘制界面，所以不要在这个函数中绘制Surface。  
\* surfaceChanged(SurfaceHolder holder, int format, int width,int height): 当Surface的状态（大小和格式）发生变化时会调用该函数，在surfaceCreated调用后该函数至少会被调用一次。

例子

```
public class MySurfaceViewAnimation extends SurfaceView implements Callback, Runnable {
    private Thread th;
    private SurfaceHolder sfh;
    private Canvas canvas;
    private Paint paint;
    private Bitmap bmp;
    private int bmp_x, bmp_y;
    private boolean himi; //备注1
    //back键之后后回到这个界面，会再次调用这个构造方法；home键不会
    public MySurfaceViewAnimation(Context context) {
        super(context);
        this.setKeepScreenOn(true);
        bmp = BitmapFactory.decodeResource(getResources(), R.drawable.himi_dream);
        sfh = this.getHolder();
        sfh.addCallback(this);
        paint = new Paint();
        paint.setAntiAlias(true);
        this.setLongClickable(true);
        Log.e("Himi", "MySurfaceViewAnimation");
    }
    public void surfaceCreated(SurfaceHolder holder) {
        himi = true;
        th = new Thread(this, "himi_Thread_one");//备注2
        th.start();
        Log.e("Himi", "surfaceCreated");
    }
    public void surfaceChanged(SurfaceHolder holder, int format, int width, int height) {
        Log.e("Himi", "surfaceChanged");
    }
    public void surfaceDestroyed(SurfaceHolder holder) {
        himi = false;//备注3
        Log.e("Himi", "surfaceDestroyed");
    }
    public void draw() {
        try {
            //一定要判定下canvas是否为空，因为当程序切入后台的时候，canvas是获取不到的！那么canvas一旦为空，提交画布这里就会出现参数异常的错误！
            canvas = sfh.lockCanvas();
            if (canvas != null) {
                canvas.drawColor(Color.WHITE);
                canvas.drawBitmap(bmp, bmp_x, bmp_y, paint);
            }
        } catch (Exception e) {
            Log.v("Himi", "draw is Error!");
        } finally {
            if (canvas != null)
                sfh.unlockCanvasAndPost(canvas);
        }
    }
    public void run() {
        while (himi) { //备注4
            draw();
            try {
                Thread.sleep(100);
            } catch (Exception ex) {
            }
        }
    }
}
```

线程不能启动两次；不能启动多个线程。上面的例子做到了  
back键之后后回到这个界面，会调用构造方法（之后是surfaceCreated, surfaceChanged），而home键之后回到这个界面不会。

1. 我们都知一个线程启动后，只要run方法执行结束，线程就销毁了，所以我增加了一个布尔值的成员变量 himi（备注1），这里可以控制我们的线程消亡的一个开关！（备注4）
2. 在启动线程之前，设置这个布尔值为ture，让线程一直运行。
3. 在view销毁时，设置这个布尔值为false，销毁当前线程！（备注3）

人物行走这个和MOMO的例子各有千秋；MOMO使用的是普通的View，但是封装得更好。不论的一张图片还是一张图片，都是将需要的图片保存到数组里面，需要画图取出相应的图片；这里作为一个整体，通过变换也能达到同样的效果。

```
import android.content.Context;
import android.content.res.Resources;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.view.KeyEvent;
import android.view.SurfaceHolder;
import android.view.SurfaceView;
```

```

import android.view.SurfaceHolder.Callback;

public class MySurfaceView extends SurfaceView implements Callback, Runnable {
    private Thread th = new Thread(this);
    private SurfaceHolder sfh;
    private int SH, SW;
    private Canvas canvas;
    private Paint p;
    private Paint p2;
    private Resources res;
    private Bitmap bmp;
    private int bmp_x = 100, bmp_y = 100;
    private boolean UP, DOWN, LEFT, RIGHT;
    private int animation_down[] = { 0, 1, 2 };
    private int animation_up[] = { 3, 4, 5 };
    private int animation_left[] = { 6, 7, 8 };
    private int animation_right[] = { 9, 10, 11 };
    private int animation_init[] = animation_down; // 默认动作向下
    private int frame_count;

    public MySurfaceView(Context context) {
        super(context);
        this.setKeepScreenOn(true);
        res = this.getResources();
        bmp = BitmapFactory.decodeResource(res, R.drawable.enemy1);
        sfh = this.getHolder();
        sfh.addCallback(this);
        p = new Paint();
        p.setColor(Color.YELLOW);
        p2 = new Paint();
        p2.setColor(Color.RED);
        p.setAntiAlias(true);
        setFocusable(true);
    }

    public void surfaceCreated(SurfaceHolder holder) {
        SH = this.getHeight();
        SW = this.getWidth();
        th.start();
    }

    public void draw() {
        canvas = sfh.lockCanvas();
        canvas.drawRect(0, 0, SW, SH, p);
        canvas.save();
        canvas.drawText("Himi", bmp_x-2, bmp_y-10, p2);

        // 什么意思? 这里的clipRect是设置可视区域, 记得要 canvas.save(); canvas.restore();
        canvas.clipRect(bmp_x, bmp_y, bmp_x + bmp.getWidth() / 13, bmp_y+bmp.getHeight());

        // 没看懂, 怎么样让图片动起来的?
        if (animation_init == animation_up) {
            canvas.drawBitmap(bmp, bmp_x - animation_up[frame_count] * (bmp.getWidth() / 13), bmp_y, p);
        } else if (animation_init == animation_down) {
            canvas.drawBitmap(bmp, bmp_x - animation_down[frame_count] * (bmp.getWidth() / 13), bmp_y, p);
        } else if (animation_init == animation_left) {
            canvas.drawBitmap(bmp, bmp_x - animation_left[frame_count] * (bmp.getWidth() / 13), bmp_y, p);
        } else if (animation_init == animation_right) {
            canvas.drawBitmap(bmp, bmp_x - animation_right[frame_count] * (bmp.getWidth() / 13), bmp_y, p);
        }
        canvas.restore();
        sfh.unlockCanvasAndPost(canvas);
    }

    public void cycle() {
        if (DOWN) {
            bmp_y += 5;
        } else if (UP) {
            bmp_y -= 5;
        } else if (LEFT) {
            bmp_x -= 5;
        } else if (RIGHT) {
            bmp_x += 5;
        }
        if (DOWN || UP || LEFT || RIGHT) {
            if (frame_count < 2) {
                frame_count++;
            } else {
                frame_count = 0;
            }
        }
        if (DOWN == false && UP == false && LEFT == false && RIGHT == false) {
            frame_count = 0; // 一松开手, 就会最终显示每个方向的第一张图片
        }
    }

    @Override
    public boolean onKeyDown(int key, KeyEvent event) {
        if (key == KeyEvent.KEYCODE_DPAD_UP) {
            if (UP == false) {
                animation_init = animation_up;
            }
            UP = true;
        } else if (key == KeyEvent.KEYCODE_DPAD_DOWN) {
            if (DOWN == false) {
                animation_init = animation_down;
            }
            DOWN = true;
        } else if (key == KeyEvent.KEYCODE_DPAD_LEFT) {
            if (LEFT == false) {
                animation_init = animation_left;
            }
            LEFT = true;
        } else if (key == KeyEvent.KEYCODE_DPAD_RIGHT) {
            if (RIGHT == false) {
                animation_init = animation_right;
            }
            RIGHT = true;
        }
        return super.onKeyDown(key, event);
    }

    @Override
    public boolean onKeyUp(int keyCode, KeyEvent event) {
        if (DOWN) {
            DOWN = false;
        } else if (UP) {
            UP = false;
        } else if (LEFT) {
            LEFT = false;
        }
    }
}

```





```

        default:
            break;
        }
    }
    list.add(map);
}

```

接下来，我们一个页面最多只包含10个书目信息，而我们校图书馆，光以“Java”为关键字的书就超过1000本，怎么来显示完全呢，一次显示所有的书肯定不现实。首先数据量太大，手机无法承受；消耗流量过大，用户体验极差。所以，我们就需要ListView能够动态加载数据，即一开始显示十项，如果用户此时拉动ListView显示完十项之后，自动联网，再加载十项(如果还有十项的话)，这样的用户体验会非常顺畅。

这个功能的核心是，我们的ListView需要实现OnScrollListener接口。

如果你的ListView所在的Activity继承的是ListActivity的话，只需在extends ListActivity后面加上implements OnScrollListener，这时你需要复写onScroll和onScrollStateChanged。如果你的ListView是从XMLgetView得到的，你只需为它setOnScrollListener，也会需要你复写onScroll和onScrollStateChanged。

不管你用哪种方法，我们只用修改onScroll方法

```

@Override
public void onScroll(AbsListView view, int firstVisibleItem, int visibleItemCount, int totalItemCount) {
    //关键的判断代码，这句话表示用户将ListView拉至最底部
    if(firstVisibleItem + visibleItemCount == totalItemCount){

    }
    //你只需要把继续得到下面十项的代码写在这里，就可以实现上述功能了。
    //同样再使用一次POST方法，不再赘述
    //代码由于同样原因不予以展示
}

```

## 14、搜索

android源码解析 ---- Mms 短信息，如何使用系统浮动搜索框

<http://blog.csdn.net/dany1202/article/details/6361157>

1.配置search bar的相关信息，位置res/xml/searchable.xml

```

<searchable xmlns:android="http://schemas.android.com/apk/res/android"
    2.     android:label="@string/search_label"
    3.     android:hint="@string/search_hint"
    4.     android:searchSettingsDescription="@string/search_setting_description"
    5.     android:searchSuggestAuthority="com.android.mms.SuggestionsProvider"
    6.     android:searchSuggestSelection="pattern" ?
    7.     android:searchSuggestIntentAction="android.intent.action.SEARCH"
    8.     android:includeInGlobalSearch="true"
    9. />

```

说明：label和hint要有，其中label默认没有显示，如果想要显示，添加：android.searchMode="showSearchLabelAsBadge" 一般不用，searchSettingsDescription为搜索设置项中短信息搜索列表项的说明；

searchSuggestAuthority为对应的权限声明，其值与manifest.xml中SuggestionsProvider类的android.authorities的值相同。includeInGlobalSearch为是否包含在全局搜索中，为true时，搜索设置项可以检索到它。

2.manifest.xml搜索结果activity配置

```

1. <activity android:name=".ui.SearchActivity"
2.     android:label="@string/search"
3.     android:configChanges="orientation|keyboardHidden" >
4.     <intent-filter>
5.         <action android:name="android.intent.action.SEARCH" />
6.         <category android:name="android.intent.category.DEFAULT" />
7.     </intent-filter>
8.     <meta-data android:name="android.app.searchable"
9.         android:resource="@xml/searchable" />
10. </activity>

```

说明：此activity用来显示搜索结果列表界面，继承自ListActivity。

3.是所有的应用程序都可以调用搜索键

```

<meta-data android:name="android.app.default_searchable"
android:value=".ui.SearchActivity" />

```

其中SearchActivity为执行并显示搜索结果的activity。

4.提供搜索功能的provider

```

1. <!-- Search Suggestions Provider -->
2. <provider android:name="SuggestionsProvider"
3.     android:readPermission="android.permission.READ_SMS"
4.     android:authorities="com.android.mms.SuggestionsProvider" >
5.     <path-permission
6.         android:pathPrefix="/search_suggest_query"
7.         android:readPermission="android.permission.GLOBAL_SEARCH" />
8.     <path-permission
9.         android:pathPrefix="/search_suggest_shortcut"
10.        android:readPermission="android.permission.GLOBAL_SEARCH" />
11. </provider>

```

5.搜索键响应函数

在当前的应用程序界面，按下搜索键时，会自动调用onSearchRequested()方法，你可以重写这个方法，做你想要的操作。如果不重写，会调用其父类startSearch(...) 如果想在某个activity中屏蔽掉搜索框，重写

```

@Override
public boolean onSearchRequested() {
    return false;
}
)即可。
或者如果你想进行一些操作
@Override
public boolean onSearchRequested() {
    doSomething();//some thing you want to do
    return super.onSearchRequested();
}

```

6.响应搜索功能

```

1. public class SearchActivity extends ListActivity{
2.     @Override
3.     public void onCreate(Bundle icle){
4.         super.onCreate(icle);
5.         setContentView(R.layout.search_activity);
6.         String searchStringParameter = getIntent().getStringExtra(SearchManager.QUERY);
7.         //做你自己的查询数据操作
8.         doMySearch(searchStringParameter);
9.         ...
10.    }
11. ....
12.}

```

7.自己写了个搜索结果界面，分享一下，仿照短信息写的

```
1. public class NoteSearchableActivity extends ListActivity{
2.
3.     private ListView searchList;
4.     private TextView mTextView;
5.     String searchString;
6.     private AsyncQueryHandler mQueryHandler;
7.     private Uri mUri;
8.
9.     @Override
10.    public void onCreate(Bundle savedInstanceState) {
11.        super.onCreate(savedInstanceState);
12.        setContentView(R.layout.search_activity);
13.
14.        mTextView = (TextView) findViewById(android.R.id.empty);
15.        mTextView.setVisibility(View.GONE);
16.
17.        searchList = getListView();
18.        searchList.setItemsCanFocus(true);
19.        searchList.setFocusable(true);
20.        searchList.setClickable(true);
21.
22.        ContentResolver cr = getContentResolver();
23.
24.        Intent intent = getIntent();
25.        if (Intent.ACTION_SEARCH.equals(intent.getAction())) {
26.            searchString = intent.getStringExtra(SearchManager.QUERY).trim();
27.        }
28.
29.        mQueryHandler = new AsyncQueryHandler(cr) {
30.            protected void onQueryComplete(int token, Object cookie, Cursor c) {
31.                if (c == null) {
32.                    mTextView.setVisibility(View.VISIBLE);
33.                    return;
34.                }
35.                int cursorCount = c.getCount();
36.
37.                setTitle(getString(R.string.search_activity_title,searchString));
38.
39.                if(cursorCount>=1){
40.                    mTextView.setVisibility(View.VISIBLE);
41.                }
42.
43.                setListAdapter(new CursorAdapter(NoteSearchableActivity.this,
44.                    c, false /* no autoquery */) {
45.
46.                    @Override
47.                    public void bindView(View view, Context context, Cursor cursor) {
48.
49.                        TextView noteTitle = (TextView) view.findViewById(R.id.note_list_title);
50.                        TextView noteData = (TextView) view.findViewById(R.id.note_list_time);
51.
52.                        noteTitle.setText(cursor.getString(3));
53.                        noteData.setText(getDate(cursor.getLong(2)));
54.
55.                        view.setOnClickListener(new View.OnClickListener() {
56.                            public void onClick(View v) {
57.                                final Intent onClickIntent = new Intent(NoteSearchableActivity.this, EditNote.class);
58.                                onClickIntent.putExtra("thread_id", threadId);
59.                                onClickIntent.putExtra("highlight", searchString);
60.                                onClickIntent.putExtra("select_id", rowId);
61.                                startActivity(onClickIntent);
62.                            }
63.                        });
64.                    }
65.
66.                    @Override
67.                    public View newView(Context context, Cursor cursor, ViewGroup parent) {
68.                        LayoutInflater inflater = LayoutInflater.from(context);
69.                        View v = inflater.inflate(R.layout.note_item, parent, false);
70.                        return v;
71.                    }
72.                });
73.
74.                // Remember the query if there are actual results
75.                if (cursorCount > 0) {
76.                    SearchRecentSuggestions recent = ((MmsApp) getApplication()).getRecentSuggestions();
77.                    if (recent != null) {
78.                        recent.saveRecentQuery(
79.                            searchString,
80.                            getString(R.string.search_history,
81.                                cursorCount, searchString));
82.                    }
83.                }
84.            }
85.        });
86.
87.        mUri = NoteProvider.CONTENT_URI.buildUpon()
88.            .appendQueryParameter("pattern", searchString).build();
89.
90.    }
91.
92.    @Override
93.    protected void onStart() {
94.        mQueryHandler.startQuery(0, null, NoteProvider.CONTENT_URI, null, "content like '%" + searchString + "%'", null, null);
95.        super.onStart();
96.    }
97.
98.
99. }
```

## 8.构建Content Provider

在QuickSearchBox中通过ContentResolver调用内容提供者的query(Uri, String[], String, String[], String)方法进行信息搜索。

应用程序必须实现这个方法并且返回搜索到的Cursor对象。

下面将对ContentProvider的query方法的参数进行说明：

第一个参数(uri)：

content://authority/suggestion.path/search\_suggest\_query /queryStr? limit=50

authority: 对应searchable.xml文件中的android:searchSuggestAuthority属性。

suggestion.path: 对应searchable.xml文件中的android:searchSuggestPath属性。



search\_suggest\_query : 固定字符串。  
queryStr: 查询字符串。  
limit : 查询条数。  
第二个参数 (projection) : 一直为null。  
第三个参数 (selection) : 对应searchable.xml文件中android:searchSuggestSelection属性,ContentProvider的query方法将会调用SQLite数据库的query方法,用这个参数组成SQL语句中where后面的条件,如: where name like ?, "name like ? "就是selection参数的值,注意, android:searchSuggestSelection属性中配置的查询条件不能用"AND"或"OR"等符号连接多个条件 (不能写成: "name like ? AND age like ?") 。  
第四个参数 (selectionArgs) : 如果searchable.xml文件中android:searchSuggestSelection属性值不为空,程序会将查询字符串做为这个数组的第一个元素,也是唯一的元素,在上面第三个参数中定义了查询条件,这个参数的作用是给查询条件中的问号赋值。  
第五个参数: 一直为null。  
该方法将搜索到的数据以cursor对象的形式返回。

上面的说法貌似有错误啊。

联系人

WebView清除缓存

1.删除保存于手机上的缓存.

复制到剪贴板 Java代码

```
1. // clear the cache before time numDays
2. private int clearCacheFolder (File dir, long numDays) {
3. int deletedFiles = 0;
4. if (dir!= null && dir.isDirectory () ) {
5. try {
6. for (File child: dir.listFiles () ) {
7. if (child.isDirectory () ) {
8. deletedFiles += clearCacheFolder (child, numDays);
9. }
10. if (child.lastModified () < numDays) {
11. if (child.delete () ) {
12. deletedFiles++;
13. }
14. }
15. }
16. } catch (Exception e) {
17. e.printStackTrace ();
18. }
19. }
20. return deletedFiles;
21. }
```

复制代码

调用: clearCacheFolder (Activity.getCacheDir (), System.currentTimeMillis () ); //删除此时之前的缓存.

2. 打开关闭使用缓存:

优先使用缓存:

1.WebView.getSettings () .setCacheMode (WebSettings.LOAD\_CACHE\_ELSE\_NETWORK);

不使用缓存:

1.WebView.getSettings () .setCacheMode (WebSettings.LOAD\_NO\_CACHE);

3.在退出应用的时候加上如下代码:

复制到剪贴板 Java代码

```
1. File file = CacheManager.getCacheFileBaseDir ();
2. if (file != null && file.exists () && file.isDirectory () ) { for (File item : file.listFiles () ) { item.delete ();
3. }
4. file.delete ();
5. }
6. context.deleteDatabase ("webview.db");
7. context.deleteDatabase ("webviewCache.db");
8.
```

复制代码

<http://www.apkbus.com/forum.php?mod=viewthread&tid=15259>

省略号的问题

<http://code.google.com/p/android/issues/detail?id=2254>

所有程序

List<ApplicationInfo> android.content.pm.PackageManager.getInstalledApplications(int flags)  
appInfos = packageManager.getInstalledApplications(PackageManager.GET\_UNINSTALLED\_PACKAGES);

删除程序: String packname = appinfo.packageName;  
Intent intent = new Intent();  
intent.setAction(Intent.ACTION\_DELETE);  
Uri data = Uri.parse("package:" + packname);  
intent.setData(data);  
startActivity(intent);

安装apk

```
Intent intent = new Intent();
intent.setAction(Intent.ACTION_VIEW);
//data uri tel:1234 context://
// intent.setData(Uri.fromFile(file));
// //type MIME 的类型 . 图片 image/jpeg audio/MP3
// intent.setType("application/vnd.android.package-archive");
intent.setDataAndType(Uri.fromFile(file), "application/vnd.android.package-archive");
startActivity(intent);
```

opickmo中的例子

```
1. Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
2. intent.setType("audio/*");
3. startActivity(Intent.createChooser(intent, "Select music"));
```

```

<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_window_focused="false"
        android:drawable="@color/transparent" />

    <!-- Even though these two point to the same resource, have two states so the drawable will invalidate itself when coming out of pressed state. -->
    <item android:state_focused="true" android:state_enabled="false"
        android:state_pressed="true"
        android:drawable="@drawable/list_selector_background_disabled" />
    <item android:state_focused="true" android:state_enabled="false"
        android:drawable="@drawable/list_selector_background_disabled" />

    <item android:state_focused="true" android:state_pressed="true"
        android:drawable="@drawable/list_selector_background_transition" />
    <item android:state_focused="false" android:state_pressed="true"
        android:drawable="@drawable/list_selector_background_transition" />

    <item android:state_focused="true"
        android:drawable="@drawable/list_selector_background_focus" />
</selector>

```

```

public static String inputStreamToString(final InputStream stream)
    throws IOException {
    BufferedReader br = new BufferedReader(new InputStreamReader(stream,
        "UTF-8"));
    StringBuilder sb = new StringBuilder();
    String line = null;
    while ((line = br.readLine()) != null) {
        sb.append(line + "\n");
    }
    br.close();
    return sb.toString();
}

private static String read(HttpResponse response) throws IOException {
    String result = "";
    HttpEntity entity = response.getEntity();
    InputStream inputStream;
    inputStream = entity.getContent();
    ByteArrayOutputStream content = new ByteArrayOutputStream();

    Header header = response.getFirstHeader("Content-Encoding");
    if (header != null
        && header.getValue().toLowerCase().indexOf("gzip") > -1) {
        inputStream = new GZIPInputStream(inputStream);
    }

    // Read response into a buffered stream
    int readBytes = 0;
    byte[] sBuffer = new byte[512];
    while ((readBytes = inputStream.read(sBuffer)) != -1) {
        content.write(sBuffer, 0, readBytes);
    }
    // Return result from buffered stream
    result = new String(content.toByteArray(), "UTF-8");
    return result;
}

URLConnection connection=new URL(url).openConnection();
InputStream stream=connection.getInputStream();
BufferedInputStream in=new BufferedInputStream(stream);
ByteArrayOutputStream out=new ByteArrayOutputStream(10240);
int read;
byte[] b=new byte[4096];

while ((read = in.read(b)) != -1) {
    out.write(b, 0, read);
}

out.flush();
out.close();

byte[] raw=out.toByteArray();

put(url, new BitmapDrawable(new ByteArrayInputStream(raw)));

M message=(M)params[0];

if (message!=null) {
    bus.send(message);
}

if (cache!=null) {
    FileOutputStream file=new FileOutputStream(cache);
    //java.io.FileNotFoundException: /mnt/sdcard/ca89ad9535a5efe65810e755fc6984 (Permission denied)
    file.write(raw);
    file.flush();
    file.close();
}

```

之前在写好Notification之后，发现按Home回到主界面，再按通知栏的消息（Notification），并没有回到退出之前正在运行的Activity，后来尝试了挺多方法总是失败。不过我最终还是解决了这个问题，主要是在代码中加入两行代码作为声名即可。废话不多说，如下：

```

Intent notificationIntent = new Intent(this.context,this.context.getClass());

/*add the following two lines to resume the app same with previous statuses*/
notificationIntent.setAction(Intent.ACTION_MAIN);
notificationIntent.addCategory(Intent.CATEGORY_LAUNCHER);
/**/
PendingIntent contentIntent = PendingIntent.getActivity(this.context, 0, notificationIntent,PendingIntent.FLAG_UPDATE_CURRENT);
notification.setLatestEventInfo(context, contentTitle, contentText, contentIntent);
mNotificationManager.notify(NOTIFICATION_SERVICE_ID,notification);

上面实验是错误的
或者
nm = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
notification = new Notification();

```

```

        updateIntent = new Intent(this, SplashActivity.class);
        updateIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_NEW_TASK);
        pi = PendingIntent.getActivity(this, 0, null, 0);
        // 设置通知栏显示内容
        notification.icon = R.drawable.setting_bind;
        notification.tickerText = "下载中";
        notification.flags |= Notification.FLAG_AUTO_CANCEL;
        notification.setLatestEventInfo(this, "果库升级", "0%", pi);

/**
 * Notification notification = new Notification(R.drawable.icon, "收到公文", System.currentTimeMillis());
 * notification.setLatestEventInfo(getApplicationContext(), "标题", "内容", PendingIntent.getActivity(LaunchNotificationActivity.this, 0, new
Intent(LaunchNotificationActivity.this, LaunchNotificationActivity.class), 0));
 * notification.flags |= Notification.FLAG_AUTO_CANCEL;
 * notification.defaults |= Notification.DEFAULT_SOUND;
 * notification.defaults |= Notification.DEFAULT_VIBRATE; //调用系统默认的声音提示以及震动加入cuses-permission android:name="android.permission.VIBRATE" /><!-- 允许震动 -->
//
 * manager.notify(1, notification);
 * @author wangliang
 *
 */

```

```

ProgressBar
达到进度条的效果，方式一：
<?xml version="1.0" encoding="utf-8"?>
<rotate xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="750000"
    android:fillEnabled="true"
    android:fromDegrees="0.0"
    android:pivotX="50.0%"
    android:pivotY="50.0%"
    android:repeatCount="infinite"
    android:toDegrees="360000"
    android:interpolator="@android:anim/linear_interpolator"
/>

```

这种方式不好，因为android:interpolator不好用，速度不一样（我加上alpha两个动画也不好）。于是progress的属性

```

<style name="Widget.ProgressBar">
    <item name="android:indeterminateOnly">true</item>
    <item name="android:indeterminateDrawable">@android:drawable/progress_medium_white</item>
    <item name="android:indeterminateBehavior">repeat</item>
    <item name="android:indeterminateDuration">3500</item>
    <item name="android:minWidth">48dp</item>
    <item name="android:maxWidth">48dp</item>
    <item name="android:minHeight">48dp</item>
    <item name="android:maxHeight">48dp</item>
</style>

```

这是android的style集合，其中progress\_medium\_white有些属性不能用，改成自己的：

```

<ProgressBar
    android:id="@+id/throbber"
    style="@style/progressStyleMain"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:indeterminateDrawable="@drawable/progress_medium_white"
/>
<style name="progressStyleMain">
    <item name="android:maxWidth">32.0dp</item>
    <item name="android:maxHeight">32.0dp</item>
    <item name="android:indeterminateDrawable">@drawable/progress_loading</item>
    <item name="android:minWidth">32.0dp</item>
    <item name="android:minHeight">32.0dp</item>
</style>

```

而progress\_loading为：

```

<?xml version="1.0" encoding="utf-8"?>
<animated-rotate xmlns:android="http://schemas.android.com/apk/res/android"
    android:drawable="@drawable/refresh_home"
    android:duration="100"
    android:fromDegrees="0.0"
    android:interpolator="@android:anim/linear_interpolator"
    android:pivotX="50.0%"
    android:pivotY="50.0%"
    android:startOffset="0"
    android:toDegrees="360.0"
/>

```

注意它是图片，不是动画（好像绿色背景的属性没有提示，而且线性插值器什么也没有）。注意的是 android:progressDrawable="@drawable/progress\_loading\_white"没有效果的，必须是 android:indeterminateDrawable属性，而且图片为动画图片才有效果。

自定义属性

```

public static final class attr {
    public static final int seg_cnt2=0x7f010000;
}

public static final class styleable {
    public static final int[] DashLineET = {0x7f010000};
    public static final int DashLineET_seg_cnt2 = 0; //
};
<declare-styleable name="DashLineET"> //styleable中数组的名字
    <attr name="seg_cnt2" format="integer" /> //styleable中属性的名字为DashLineET_seg_cnt2
</declare-styleable>

```

```

private void init(Context context, AttributeSet attrs) {
    // .....
    TypedArray a = context.obtainStyledAttributes(attrs, R.styleable.DashLineET);
    SEG_CNT = a.getInt(R.styleable.DashLineET_seg_cnt2, SEG_CNT);
}

```

可以这样使用

```

<style name="mytest">
    <item name="seg_cnt2">18</item>
</style>

```

也可以这样

```

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:heima="http://schemas.android.com/apk/res/heima.droid"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <heima.droid.DashLineEditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="rtt"
        heima:seg_cnt2="3"
    />

```

```

        android:background="#00f" />
</LinearLayout>
注意前者没有命名空间

```

在声明**Notification**的跳转**Intent**时，需要给其添加上述红色标出的两行代码，即可使每次按**Notification**时回到原先正在运行的**Activity**上面。希望对大家有帮助

```

    }
}

```

图片：  
 android:src="@android:drawable/progress\_horizontal"，具体内容：

```

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

<layer-list xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:id="@android:id/background">
        <shape>
            <corners android:radius="5dip" />
            <gradient
                android:startColor="#ff9d9e9d"
                android:centerColor="#ff5a5d5a"
                android:centerY="0.75"
                android:endColor="#ff747674"
                android:angle="270"
            />
        </shape>
    </item>

    <item android:id="@android:id/secondaryProgress">
        <clip>
            <shape>
                <corners android:radius="5dip" />
                <gradient
                    android:startColor="#80ffd300"
                    android:centerColor="#80ffb600"
                    android:centerY="0.75"
                    android:endColor="#a0ffcb00"
                    android:angle="270"
                />
            </shape>
        </clip>
    </item>

    <item android:id="@android:id/progress">
        <clip>
            <shape>
                <corners android:radius="5dip" />
                <gradient
                    android:startColor="#ffffd300"
                    android:centerColor="#ffffb600"
                    android:centerY="0.75"
                    android:endColor="#ffffcb00"
                    android:angle="270"
                />
            </shape>
        </clip>
    </item>

</layer-list>

```

config文件

```

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

<!-- These resources are around just to allow their values to be customized
     for different hardware and product builds. -->
<resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2">

    <!-- Do not translate. Defines the slots for the right-hand side icons. That is to say, the
         icons in the status bar that are not notifications. -->
    <string-array name="config_statusBarIcons">
        <item><xliff:g id="id">ime</xliff:g></item>
        <item><xliff:g id="id">sync_failing</xliff:g></item>
        <item><xliff:g id="id">sync_active</xliff:g></item>
        <item><xliff:g id="id">gps</xliff:g></item>
        <item><xliff:g id="id">bluetooth</xliff:g></item>
        <item><xliff:g id="id">nfc</xliff:g></item>
        <item><xliff:g id="id">tty</xliff:g></item>
        <item><xliff:g id="id">speakerphone</xliff:g></item>
        <item><xliff:g id="id">mute</xliff:g></item>
        <item><xliff:g id="id">volume</xliff:g></item>
        <item><xliff:g id="id">wifi</xliff:g></item>
        <item><xliff:g id="id">cdma_eri</xliff:g></item>
        <item><xliff:g id="id">data_connection</xliff:g></item>
        <item><xliff:g id="id">phone_evdo_signal</xliff:g></item>
        <item><xliff:g id="id">phone_signal</xliff:g></item>
        <item><xliff:g id="id">battery</xliff:g></item>
        <item><xliff:g id="id">alarm_clock</xliff:g></item>
        <item><xliff:g id="id">secure</xliff:g></item>
        <item><xliff:g id="id">clock</xliff:g></item>
    </string-array>

    <!-- Flag indicating whether the surface flinger has limited
         alpha compositing functionality in hardware. If set, the window
         manager will disable alpha transformation in animations where not
         strictly needed. -->
    <bool name="config_sf_limitedAlpha">false</bool>

    <!-- Default value used to block data calls if ims is not
         connected. If you use the ims app DCT will block
         any other app from connecting until ims app is connected-->
    <bool name="ImsConnectedDefaultValue">false</bool>

    <!-- Flag indicating whether the surface flinger is inefficient
         at performing a blur. Used by parts of the UI to turn off
         the blur effect where it isn't worth the performance hit.
         As of Honeycomb, blurring is not supported anymore. -->
    <bool name="config_sf_slowBlur">true</bool>

    <!-- The duration (in milliseconds) of a short animation. -->
    <integer name="config_shortAnimTime">200</integer>

    <!-- The duration (in milliseconds) of a medium-length animation. -->
    <integer name="config_mediumAnimTime">400</integer>

    <!-- The duration (in milliseconds) of a long animation. -->
    <integer name="config_longAnimTime">500</integer>

```

```

<!-- The duration (in milliseconds) of the activity open/close and fragment open/close animations. -->
<integer name="config_activityShortDur">150</integer>
<integer name="config_activityDefaultDur">220</integer>

<!-- Duration for the dim animation behind a dialog. This may be either
    a percentage, which is relative to the duration of the enter/open
    animation of the window being shown that is dimming behind, or it may
    be an integer for a constant duration. -->
<fraction name="config_dimBehindFadeDuration">100%</fraction>

<!-- The maximum width we would prefer dialogs to be. 0 if there is no
    maximum (let them grow as large as the screen). Actual values are
    specified for -large and -xlarge configurations. --&;
<dimen name="config_prefDialogWidth">320dp</dimen>

<!-- Enables or disables fading edges when marquee is enabled in TextView. -->
<bool name="config_ui_enableFadingMarquee">true</bool>

<!-- Whether dialogs should close automatically when the user touches outside
    of them. This should not normally be modified. -->
<bool name="config_closeDialogWhenTouchOutside">true</bool>

<!-- The duration (in milliseconds) that the radio will scan for a signal
    when there's no network connection. If the scan doesn't timeout, use zero -->
<integer name="config_radioScanningTimeout">0</integer>

<!-- XXXXX NOTE THE FOLLOWING RESOURCES USE THE WRONG NAMING CONVENTION.
    Please don't copy them, copy anything else. -->

<!-- This string array should be overridden by the device to present a list of network
    attributes. This is used by the connectivity manager to decide which networks can coexist
    based on the hardware -->
<!-- An Array of "[Connection name],[ConnectivityManager.TYPE_xxxx],
    [associated radio-type],[priority],[restoral-timer(ms)],[dependencyMet]  -->
<!-- the 5th element "resore-time" indicates the number of milliseconds to delay
    before automatically restore the default connection. Set -1 if the connection
    does not require auto-restore. -->
<!-- the 6th element indicates boot-time dependency-net value. -->
<string-array translatable="false" name="networkAttributes">
    <item>"wifi,1,1,1,-1,true"</item>
    <item>"mobile,0,0,0,-1,true"</item>
    <item>"mobile_mms,2,0,2,60000,true"</item>
    <item>"mobile_supl,3,0,2,60000,true"</item>
    <item>"mobile_hipri,5,0,3,60000,true"</item>
    <item>"mobile_fota,10,0,2,60000,true"</item>
    <item>"mobile_ims,11,0,2,60000,true"</item>
    <item>"mobile_cbs,12,0,2,60000,true"</item>
    <item>"wifi_p2p,13,1,0,-1,true"</item>
</string-array>

<!-- Array of ConnectivityManager.TYPE_xxxx constants for networks that may only
    be controlled by system@signature apps. -->
<integer-array translatable="false" name="config_protectedNetworks">
    <item>10</item>
    <item>11</item>
    <item>12</item>
</integer-array>

<!-- This string array should be overridden by the device to present a list of radio
    attributes. This is used by the connectivity manager to decide which networks can coexist
    based on the hardware -->
<!-- An Array of "[ConnectivityManager.connectionType],
    [# simultaneous connection types]" -->
<string-array translatable="false" name="radioAttributes">
    <item>"1,1"</item>
    <item>"0,1"</item>
</string-array>

<!-- Set of NetworkInfo.getType() that reflect data usage. -->
<integer-array translatable="false" name="config_data_usage_network_types">
    <item>0</item> <!-- TYPE_MOBILE -->
    <item>2</item> <!-- TYPE_MOBILE_MMS -->
    <item>3</item> <!-- TYPE_MOBILE_SUPL -->
    <item>4</item> <!-- TYPE_MOBILE_DUN -->
    <item>5</item> <!-- TYPE_MOBILE_HIPRI -->
    <item>10</item> <!-- TYPE_MOBILE_FOTA -->
    <item>11</item> <!-- TYPE_MOBILE_IMS -->
    <item>12</item> <!-- TYPE_MOBILE_CBS -->
</integer-array>

<!-- The maximum duration (in milliseconds) we expect a network transition to take -->
<integer name="config_networkTransitionTimeout">60000</integer>

<!-- List of regexexpressions describing the interface (if any) that represent tetherable
    USB interfaces. If the device doesn't want to support tethering over USB this should
    be empty. An example would be "usb.*" -->
<string-array translatable="false" name="config_tether_usb_regexs">
</string-array>

<!-- List of regexexpressions describing the interface (if any) that represent tetherable
    Wifi interfaces. If the device doesn't want to support tethering over Wifi this
    should be empty. An example would be "softap.*" -->
<string-array translatable="false" name="config_tether_wifi_regexs">
</string-array>

<!-- List of regexexpressions describing the interface (if any) that represent tetherable
    Bluetooth interfaces. If the device doesn't want to support tethering over Bluetooth this
    should be empty. -->
<string-array translatable="false" name="config_tether_bluetooth_regexs">
</string-array>

<!-- Max number of Bluetooth tethering connections allowed. If this is
    updated config_tether_dhcp_range has to be updated appropriately. -->
<integer translatable="false" name="config_max_p2p_devices">5</integer>

<!-- Dhcp range (min, max) to use for tethering purposes -->
<string-array translatable="false" name="config_tether_dhcp_range">
</string-array>

<!-- Regex of wired ethernet ifaces -->
<string translatable="false" name="config_ethernet_iface_regex">eth\d</string>

<!-- If the mobile hotspot feature requires provisioning, an intent string can be provided
    to the launch a supported application that provisions the devices.

    Example Usage:

    Intent intent = new Intent(R.string.config_mobile_hotspot_provision_intent);
    startActivityForResult(intent, 0);

```

```

    public void onActivityResult(int requestCode, int resultCode, Intent intent) {
        super.onActivityResult(requestCode, resultCode, intent);
        if (requestCode == 0) {
            if (resultCode == Activity.RESULT_OK) {
                //Mobile hotspot provisioning successful
            } else {
                //Mobile hotspot provisioning failed
            }
        }
    }

    See src/com/android/settings/TetherSettings.java for more details.
    -->
    <string translatable="false" name="config_mobile_hotspot_provision_intent"></string>

    <!-- Array of ConnectivityManager.TYPE_xxxx values allowable for tethering -->
    <!-- Common options are [1, 4] for TYPE_WIFI and TYPE_MOBILE_DUN or
    <!-- [0,1,5,7] for TYPE_MOBILE, TYPE_WIFI, TYPE_MOBILE_HIPRI and TYPE_BLUETOOTH -->
    <integer-array translatable="false" name="config_tether_upstream_types">
        <item>1</item>
        <item>4</item>
    </integer-array>

    <!-- String containing the apn value for tethering. May be overridden by secure settings
    TETHER_DUN_APN. Value is a comma separated series of strings:
    "name,apn,proxy,port,username,password,server,mmsc,mmsproxy,mmsport,mcc,mnc,auth,type"
    note that empty fields can be omitted: "name,apn,,,,,,,,,310,260,,DUN" -->
    <string translatable="false" name="config_tether_apndata"></string>

    <!-- Boolean indicating whether the wifi chipset has dual frequency band support -->
    <bool translatable="false" name="config_wifi_dual_band_support">false</bool>

    <!-- Boolean indicating whether the wifi chipset has p2p support -->
    <bool translatable="false" name="config_wifi_p2p_support">false</bool>
    <!-- Device type information conforming to Annex B format in WiFi Direct specification.
    The default represents a dual-mode smartphone -->
    <string translatable="false" name="config_wifi_p2p_device_type">10-0050F204-5</string>

    <!-- Boolean indicating whether the wifi chipset supports background scanning mechanism.
    This mechanism allows the host to remain in suspend state and the dongle to actively
    scan and wake the host when a configured SSID is detected by the dongle. This chipset
    capability can provide power savings when wifi needs to be always kept on.
    The driver commands needed to support the feature are BGSCAN-START and BGSCAN-STOP -->
    <bool translatable="false" name="config_wifi_background_scan_support">false</bool>

    <!-- Integer indicating wpa_supplicant scan interval in milliseconds -->
    <integer translatable="false" name="config_wifi_supplicant_scan_interval">15000</integer>

    <!-- Integer indicating the framework scan interval in milliseconds. This is used in the scenario
    where the chipset does not support background scanning (Config_wifi_background_scan_support
    is false) to set up a periodic wake up scan so that the device can connect to a new access
    point on the move. A value of 0 means no periodic scans will be used in the framework. -->
    <integer translatable="false" name="config_wifi_framework_scan_interval">300000</integer>

    <!-- Flag indicating whether the keyguard should be bypassed when
    the slider is open. This can be set or unset depending how easily
    the slider can be opened (for example, in a pocket or purse). -->
    <bool name="config_bypass_keyguard_if_slider_open">true</bool>

    <!-- Flag indicating whether we should enable the automatic brightness in Settings.
    Software implementation will be used if config_hardware_auto_brightness_available is not set -->
    <bool name="config_automatic_brightness_available">false</bool>

    <!-- Don't name config resources like this. It should look like config_annoyDianne -->
    <bool name="config_annoy_dianne">true</bool>

    <!-- If this is true, the screen will come on when you unplug usb/power/whatever. -->
    <bool name="config_unplugTurnsOnScreen">false</bool>

    <!-- If this is true, the screen will fade off. -->
    <bool name="config_animateScreenLights">true</bool>

    <!-- XXXXXX END OF RESOURCES USING WRONG NAMING CONVENTION -->

    <!-- If true, the screen can be rotated via the accelerometer in all 4
    rotations as the default behavior. -->
    <bool name="config_allowAllRotations">false</bool>

    <!-- If true, the direction rotation is applied to get to an application's requested
    orientation is reversed. Normally, the model is that landscape is
    clockwise from portrait; thus on a portrait device an app requesting
    landscape will cause a clockwise rotation, and on a landscape device an
    app requesting portrait will cause a counter-clockwise rotation. Setting
    true here reverses that logic. -->
    <bool name="config_reverseDefaultRotation">false</bool>

    <!-- The number of degrees to rotate the display when the keyboard is open. -->
    <integer name="config_lidOpenRotation">90</integer>

    <!-- The number of degrees to rotate the display when the device is in a desk dock.
    A value of -1 means no change in orientation by default. -->
    <integer name="config_deskDockRotation">-1</integer>

    <!-- The number of degrees to rotate the display when the device is in a car dock.
    A value of -1 means no change in orientation by default. -->
    <integer name="config_carDockRotation">-1</integer>

    <!-- Control whether being in the desk dock (and powered) always
    keeps the screen on. By default it stays on when plugged in to
    AC. 0 will not keep it on; or together 1 to stay on when plugged
    in to AC and 2 to stay on when plugged in to USB. (So 3 for both.) -->
    <integer name="config_deskDockKeepsScreenOn">1</integer>

    <!-- Control whether being in the car dock (and powered) always
    keeps the screen on. By default it stays on when plugged in to
    AC. 0 will not keep it on; or together 1 to stay on when plugged
    in to AC and 2 to stay on when plugged in to USB. (So 3 for both.) -->
    <integer name="config_carDockKeepsScreenOn">1</integer>

    <!-- Control whether being in the desk dock should enable accelerometer
    based screen orientation. Note this should probably default to true
    like car dock, but we haven't had a chance to test it. -->
    <bool name="config_deskDockEnablesAccelerometer">false</bool>

    <!-- Control whether being in the car dock should enable accelerometer based
    screen orientation. This defaults to true because putting a device in
    a car dock make the accelerometer more a physical input (like a lid). -->
    <bool name="config_carDockEnablesAccelerometer">true</bool>

    <!-- Indicate whether the lid state impacts the accessibility of

```

```

        the physical keyboard. 0 means it doesn't, 1 means it is accessible
        when the lid is open, 2 means it is accessible when the lid is
        closed. The default is 1. -->
<integer name="config_lidKeyboardAccessibility">1</integer>

<!-- Indicate whether the lid state impacts the accessibility of
        the physical keyboard. 0 means it doesn't, 1 means it is accessible
        when the lid is open, 2 means it is accessible when the lid is
        closed. The default is 0. -->
<integer name="config_lidNavigationAccessibility">0</integer>

<!-- Control the behavior when the user long presses the power button.
        0 - Nothing
        1 - Global actions menu
        2 - Power off (with confirmation)
-->
<integer name="config_longPressOnPowerBehavior">1</integer>

<!-- Indicate whether the SD card is accessible without removing the battery. -->
<bool name="config_batterySdCardAccessibility">false</bool>

<!-- List of file paths for USB host busses to exclude from USB host support.
        For example, if the first USB bus on the device is used to communicate
        with the modem or some other restricted hardware, add "/dev/bus/usb/001/"
        to this list. If this is empty, no parts of the host USB bus will be excluded.
-->
<string-array name="config_usbHostBlacklist">
</string-array>

<!-- Vibrator pattern for feedback about a long screen/key press -->
<integer-array name="config_longPressVibePattern">
    <item>0</item>
    <item>1</item>
    <item>20</item>
    <item>21</item>
</integer-array>

<!-- Vibrator pattern for feedback about touching a virtual key -->
<integer-array name="config_virtualKeyVibePattern">
    <item>0</item>
    <item>10</item>
    <item>20</item>
    <item>30</item>
</integer-array>

<!-- Vibrator pattern for a very short but reliable vibration for soft keyboard tap -->
<integer-array name="config_keyboardTapVibePattern">
    <item>40</item>
</integer-array>

<!-- Vibrator pattern for feedback about booting with safe mode disabled -->
<integer-array name="config_safeModeDisabledVibePattern">
    <item>0</item>
    <item>1</item>
    <item>20</item>
    <item>21</item>
</integer-array>

<!-- Vibrator pattern for feedback about booting with safe mode disabled -->
<integer-array name="config_safeModeEnabledVibePattern">
    <item>0</item>
    <item>1</item>
    <item>20</item>
    <item>21</item>
    <item>500</item>
    <item>600</item>
</integer-array>

<!-- Vibrator pattern for feedback about hitting a scroll barrier -->
<integer-array name="config_scrollBarrierVibePattern">
    <item>0</item>
    <item>15</item>
    <item>10</item>
    <item>10</item>
</integer-array>

<bool name="config_use_strict_phone_number_comparation">false</bool>

<!-- Display low battery warning when battery level dips to this value.
        Also, the battery stats are flushed to disk when we hit this level. -->
<integer name="config_criticalBatteryWarningLevel">4</integer>

<!-- Display low battery warning when battery level dips to this value -->
<!-- Display low battery warning when battery level dips to this value -->
<integer name="config_lowBatteryWarningLevel">15</integer>

<!-- Close low battery warning when battery level reaches this value -->
<integer name="config_lowBatteryCloseWarningLevel">20</integer>

<!-- Default color for notification LED. -->
<color name="config_defaultNotificationColor">#ff00ff00</color>

<!-- Default LED on time for notification LED in milliseconds. -->
<integer name="config_defaultNotificationLedOn">500</integer>

<!-- Default LED off time for notification LED in milliseconds. -->
<integer name="config_defaultNotificationLedOff">2000</integer>

<!-- Default value for led color when battery is low on charge -->
<integer name="config_notificationsBatteryLowARGB">0xFFFF0000</integer>

<!-- Default value for led color when battery is medium charged -->
<integer name="config_notificationsBatteryMediumARGB">0xFFFFFFFF</integer>

<!-- Default value for led color when battery is fully charged -->
<integer name="config_notificationsBatteryFullARGB">0xFF00FF00</integer>

<!-- Default value for LED on time when the battery is low on charge in milliseconds -->
<integer name="config_notificationsBatteryLedOn">125</integer>

<!-- Is the notification LED intrusive? Used to decide if there should be a disable option -->
<bool name="config_intrusiveNotificationLed">false</bool>

<!-- Default value for LED off time when the battery is low on charge in milliseconds -->
<integer name="config_notificationsBatteryLedOff">2875</integer>

<!-- Allow the menu hard key to be disabled in LockScreen on some devices -->
<bool name="config_disableMenuKeyInLockScreen">false</bool>

<!-- Don't show lock screen before unlock screen (PIN/pattern/password) -->

```

```

<bool name="config_enableLockBeforeUnlockScreen">false</bool>

<!-- Diabie lockscreen rotation by default -->
<bool name="config_enableLockScreenRotation">false</bool>

<!-- Enable puk unlockscreen by default.
      If unlock screen is disabled, the puk should be unlocked through Emergency Dialer -->
<bool name="config_enable_puk_unlock_screen">true</bool>

<!-- Control the behavior when the user long presses the home button.
      0 - Nothing
      1 - Recent apps dialog
      2 - Recent apps view in SystemUI
      This needs to match the constants in
      policy/src/com/android/internal/policy/impl/PhoneWindowManager.java
-->
<integer name="config_longPressOnHomeBehavior">2</integer>

<!-- Array of light sensor LUX values to define our levels for auto backlight brightness support.
      The N entries of this array define N + 1 zones as follows:

      Zone 0:      0 <= LUX < array[0]
      Zone 1:      array[0] <= LUX < array[1]
      ...
      Zone N:      array[N - 1] <= LUX < array[N]
      Zone N + 1:  array[N] <= LUX < infinity

      Must be overridden in platform specific overlays -->
<integer-array name="config_autoBrightnessLevels">
</integer-array>

<!-- Minimum screen brightness allowed by the power manager. -->
<integer name="config_screenBrightnessDim">20</integer>

<!-- Array of output values for LCD backlight corresponding to the LUX values
      in the config_autoBrightnessLevels array. This array should have size one greater
      than the size of the config_autoBrightnessLevels array.
      This must be overridden in platform specific overlays -->
<integer-array name="config_autoBrightnessLcdBacklightValues">
</integer-array>

<!-- Array of output values for button backlight corresponding to the LUX values
      in the config_autoBrightnessLevels array. This array should have size one greater
      than the size of the config_autoBrightnessLevels array.
      This must be overridden in platform specific overlays -->
<integer-array name="config_autoBrightnessButtonBacklightValues">
</integer-array>

<!-- Array of output values for keyboard backlight corresponding to the LUX values
      in the config_autoBrightnessLevels array. This array should have size one greater
      than the size of the config_autoBrightnessLevels array.
      This must be overridden in platform specific overlays -->
<integer-array name="config_autoBrightnessKeyboardBacklightValues">
</integer-array>

<!-- Amount of time it takes for the light sensor to warm up in milliseconds.
      For this time after the screen turns on, the Power Manager
      will not debounce light sensor readings -->
<integer name="config_lightSensorWarmupTime">0</integer>

<!-- Enables swipe versus poly-finger touch disambiguation in the KeyboardView -->
<bool name="config_swipeDisambiguation">true</bool>

<!-- Specifies the amount of time to disable virtual keys after the screen is touched
      in order to filter out accidental virtual key presses due to swiping gestures
      or taps near the edge of the display. May be 0 to disable the feature.
      It is recommended that this value be no more than 250 ms.
      This feature should be disabled for most devices. -->
<integer name="config_virtualKeyQuietTimeMillis">0</integer>

<!-- Component name of the default wallpaper. This will be ImageWallpaper if not
      specified -->
<string name="default_wallpaper_component">@null</string>

<!-- Component name of the service providing network location support. -->
<string name="config_networkLocationProvider">@null</string>

<!-- Component name of the service providing geocoder API support. -->
<string name="config_geocodeProvider">@null</string>

<!-- Boolean indicating if current platform supports bluetooth SCO for off call
      use cases -->
<bool name="config_bluetooth_sco_off_call">true</bool>

<!-- Boolean indicating if current platform supports bluetooth wide band
      speech -->
<bool name="config_bluetooth_wide_band_speech">true</bool>

<!-- Boolean indicating if current platform supports quick switch-on/off of
      Bluetooth Module -->
<bool name="config_bluetooth_adapter_quick_switch">true</bool>

<!-- The default data-use polling period. -->
<integer name="config_datause_polling_period_sec">600</integer>

<!-- The default data-use threshold in bytes. 0 disables-->
<integer name="config_datause_threshold_bytes">0</integer>

<!-- The default reduced-datarate value in kilobits per sec -->
<integer name="config_datause_throttle_kbitsps">300</integer>

<!-- The default iface on which to monitor data use -->
<string name="config_datause_iface">rmnet0</string>

<!-- The default reduced-datarate notification mask -->
<!-- 2 means give warning -->
<integer name="config_datause_notification_type">2</integer>

<!-- Flag indicating whether the current device is "voice capable".
      If true, this means that the device supports circuit-switched
      (i.e. voice) phone calls over the telephony network, and is
      allowed to display the in-call UI while a cellular voice call is
      active. This can be overridden to false for "data only" devices
      which can't make voice calls and don't support any in-call UI.

      Note: this flag is subtly different from the
      PackageManager.FEATURE_TELEPHONY system feature, which is
      available on *any* device with a telephony radio, even if the
      device is data-only. -->

```



```

<bool name="config_voice_capable">true</bool>

<!-- Flag indicating whether the current device allows sms service.
If true, this means that the device supports both sending and
receiving sms via the telephony network.
This can be overridden to false for "data only" devices
which can't send and receive sms message.

Note: Disable SMS also disable voicemail waiting sms,
cell broadcasting sms, and MMS. -->
<bool name="config_sms_capable">true</bool>

<!-- IP address of the dns server to use if nobody else suggests one -->
<string name="config_default_dns_server">8.8.8.8</string>

<!-- The default character set for GsmAlphabet -->
<!-- Empty string means MBCS is not considered -->
<string name="gsm_alphabet_default_charset"></string>

<!-- Enables SIP on WIFI only -->
<bool name="config_sip_wifi_only">false</bool>

<!-- Enables built-in SIP phone capability -->
<bool name="config_built_in_sip_phone">true</bool>

<!-- Boolean indicating if restoring network selection should be skipped -->
<!-- The restoring is handled by modem if it is true-->
<bool translatable="false" name="skip_restoring_network_selection">false</bool>

<!-- Number of database connections opened and managed by framework layer
to handle queries on each database. -->
<integer name="db_connection_pool_size">1</integer>

<!-- Max space (in MB) allocated to DownloadManager to store the downloaded
files if they are to be stored in DownloadManager's data dir,
which typically is /data/data/com.android.providers.downloads/files -->
<integer name="config_downloadDataDirSize">100</integer>

<!-- Max number of downloads allowed to proceed concurrently -->
<integer name="config_MaxConcurrentDownloadsAllowed">5</integer>

<!-- When the free space available in DownloadManager's data dir falls
below the percentage value specified by this param, DownloadManager
starts removing files to try to make percentage of available
free space above this threshold value. -->
<integer name="config_downloadDataDirLowSpaceThreshold">10</integer>

<!-- The URL that should be sent in an x-wap-profile header with an HTTP request,
as defined in the Open Mobile Alliance User Agent Profile specification
OMA-TS-UAPProf-V2.0-20060206-A Section 8.1.1.1. If the URL contains a 'xs'
format string then that substring will be replaced with the value of
Build.MODEL. The format string shall not be escaped. -->
<string name="config_useragentprofile_url"></string>

<!-- When a database query is executed, the results returned are paginated
in pages of size (in KB) indicated by this value -->
<integer name="config_cursorWindowSize">2048</integer>

<!-- Sets whether menu shortcuts should be displayed on panel menus when
a keyboard is present. -->
<bool name="config_showMenuShortcutsWhenKeyboardPresent">false</bool>

<!-- Do not translate. Defines the slots is Two Digit Number for dialing normally not USSD -->
<string-array name="config_twoDigitNumberPattern">
</string-array>

<!-- The VoiceMail default value is displayed to my own number if it is true -->
<bool name="config_telephony_use_own_number_for_voicemail">false</bool>

<!-- If this value is true, Sms encoded as octet is decoded by utf8 decoder.
If false, decoded by latin decoder. -->
<bool name="config_sms_utf8_support">false</bool>

<!-- If this value is true, The mms content-disposition field is supported correctly.
If false, Content-disposition fragments are ignored -->
<bool name="config_mms_content_disposition_support">true</bool>

<!-- National Language Identifier codes for the following two config items.
(from 3GPP TS 23.038 V9.1.1 Table 6.2.1.2.4.1):
0 - reserved
1 - Turkish
2 - Spanish (single shift table only)
3 - Portuguese
4 - Bengali
5 - Gujarati
6 - Hindi
7 - Kannada
8 - Malayalam
9 - Oriya
10 - Punjabi
11 - Tamil
12 - Telugu
13 - Urdu
14+ - reserved -->

<!-- National language single shift tables to enable for SMS encoding.
Decoding is always enabled. 3GPP TS 23.038 states that this feature
should not be enabled until a formal request is issued by the relevant
national regulatory body. Array elements are codes from the table above.
Example 1: devices sold in Turkey must include table 1 to conform with
By-Law Number 27230. (http://www.btk.gov.tr/eng/pdf/2009/BY-LAW_SMS.pdf)
Example 2: devices sold in India should include tables 4 through 13
to enable use of the new Release 9 tables for Indic languages. -->
<integer-array name="config_sms_enabled_single_shift_tables"></integer-array>

<!-- National language locking shift tables to enable for SMS encoding.
Decoding is always enabled. 3GPP TS 23.038 states that this feature
should not be enabled until a formal request is issued by the relevant
national regulatory body. Array elements are codes from the table above.
Example 1: devices sold in Turkey must include table 1 after the
Turkish Telecommunication Authority requires locking shift encoding
to be enabled (est. July 2012). (http://www.btk.gov.tr/eng/pdf/2009/BY-LAW_SMS.pdf)
See also: http://www.mobitech.com.tr/ersanotzurblog_en/index.php?entry=entry090223-160014
Example 2: devices sold in India should include tables 4 through 13
to enable use of the new Release 9 tables for Indic languages. -->
<integer-array name="config_sms_enabled_locking_shift_tables"></integer-array>

<!-- Set to true if the RSSI should always display CDMA signal strength even on EVDO -->
<bool name="config_alwaysUseCdmaRssi">false</bool>

```

```

<!-- If this value is true, duplicate Source/Destination port fields
in WDP header of some carriers OMADM wap push are supported.
ex: MSGTYPE-TotalSegments-CurrentSegment
-SourcePortDestPort-SourcePortDestPort-OMADM PDU
If false, not supported. -->
<bool name="config_duplicate_port_omadm_wappush">false</bool>

<!-- Maximum numerical value that will be shown in a status bar
notification icon or in the notification itself. Will be replaced
with @string/status_bar_notification_info_overflow when shown in the
UI. -->
<integer name="status_bar_notification_info_maxnum">999</integer>

<!-- Path to an ISO image to be shared with via USB mass storage.
This is intended to allow packaging drivers or tools for installation on a PC. -->
<string translatable="false" name="config_isoImagePath"></string>

<!-- Whether a software navigation bar should be shown. NOTE: in the future this may be
autodetected from the Configuration. -->
<bool name="config_showNavigationBar">false</bool>

<!-- Whether action menu items should be displayed in ALLCAPS or not.
Defaults to true. If this is not appropriate for specific locales
it should be disabled in that locale's resources. -->
<bool name="config_actionMenuItemAllCaps">true</bool>

<!-- Whether action menu items should obey the "withText" showAsAction
flag. This may be set to false for situations where space is
extremely limited. -->
<bool name="config_allowActionMenuItemTextWithIcon">false</bool>

<!-- Remote server that can provide NTP responses. -->
<string translatable="false" name="config_ntpServer">pool.ntp.org</string>
<!-- Timeout to wait for NTP server response. -->
<integer name="config_ntpTimeout">20000</integer>

<!-- Default network policy warning threshold, in megabytes. -->
<integer name="config_networkPolicyDefaultWarning">2048</integer>
</resources>

```

"@android:integer/config\_longAnimTime"

```

progressbar@style
style="@android:attr/progressbarStyleHorizontal"
<item name="progressbarStyleHorizontal">@android:style/Widget.ProgressBar.Horizontal</item> 注意name没有前缀
<style name="Widget.ProgressBar.Horizontal">
    <item name="android:indeterminateOnly">false</item>
    <item name="android:progressDrawable">@android:drawable/progress_horizontal</item>
    <item name="android:indeterminateDrawable">@android:drawable/progress_indeterminate_horizontal</item>
    <item name="android:minHeight">20dip</item>
    <item name="android:maxHeight">20dip</item>
</style>

```

其中progress\_horizontal为

```

<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@android:id/background">
        <shape>
            <corners android:radius="5dip" />
            <gradient
                android:startColor="#ff9d9e9d"
                android:centerColor="#ff5a5d5a"
                android:centerY="0.75"
                android:endColor="#ff747674"
                android:angle="270"
            />
        </shape>
    </item>
    <item android:id="@android:id/secondaryProgress">
        <clip>
            <shape>
                <corners android:radius="5dip" />
                <gradient
                    android:startColor="#00ffdb00"
                    android:centerColor="#00ffb600"
                    android:centerY="0.75"
                    android:endColor="#a0ffcb00"
                    android:angle="270"
                />
            </shape>
        </clip>
    </item>
    <item android:id="@android:id/progress">
        <clip>
            <shape>
                <corners android:radius="5dip" />
                <gradient
                    android:startColor="#ffffdb00"
                    android:centerColor="#ffffb600"
                    android:centerY="0.75"
                    android:endColor="#ffffcb00"
                    android:angle="270"
                />
            </shape>
        </clip>
    </item>
</layer-list>

```

而progress\_indeterminate\_horizontal 为

```

<animation-list
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot="false">
    <item android:drawable="@drawable/progressbar_indeterminate1" android:duration="200" />
    <item android:drawable="@drawable/progressbar_indeterminate2" android:duration="200" />
    <item android:drawable="@drawable/progressbar_indeterminate3" android:duration="200" />
</animation-list>

```

非水平的progressbar

```

<style name="Widget.ProgressBar">
    <item name="android:indeterminateOnly">true</item>
    <item name="android:indeterminateDrawable">@android:drawable/progress_medium_white</item>
    <item name="android:indeterminateBehavior">repeat</item>
    <item name="android:indeterminateDuration">3500</item>
    <item name="android:minWidth">48dip</item>
    <item name="android:maxWidth">48dip</item>
    <item name="android:minHeight">48dip</item>
    <item name="android:maxHeight">48dip</item>
</style>

```

而对于旋转的progressbar

```

<style name="Widget.ProgressBar">
    <item name="android:indeterminateOnly">true</item>
    <item name="android:indeterminateDrawable">@android:drawable/progress_medium_white</item>
    <item name="android:indeterminateBehavior">repeat</item>
    <item name="android:indeterminateDuration">3500</item>
    <item name="android:minWidth">48dip</item>
    <item name="android:maxWidth">48dip</item>
    <item name="android:minHeight">48dip</item>
    <item name="android:maxHeight">48dip</item>
</style>

```

<http://yanweimin7.iteye.com/blog/1002262>

```

<!-- <declare-styleable name="MyProgressBar">
    Defines the maximum value the progress can take.
    <attr name="max" format="integer" />
    Defines the default progress value, between 0 and max.
    <attr name="progress" format="integer" />
    Defines the secondary progress value, between 0 and max. This progress is drawn between
    the primary progress and the background. It can be ideal for media scenarios such as
    showing the buffering progress while the default progress shows the play progress.
    <attr name="secondaryProgress" format="integer" />
    Allows to enable the indeterminate mode. In this mode the progress
    bar plays an infinite looping animation.
    <attr name="indeterminate" format="boolean" />
    Restricts to ONLY indeterminate mode (state-keeping progress mode will not work).
    <attr name="indeterminateOnly" format="boolean" />
    Drawable used for the indeterminate mode.
    <attr name="indeterminateDrawable" format="reference" />
    Drawable used for the progress mode.
    <attr name="progressDrawable" format="reference" />
    Duration of the indeterminate animation.
    <attr name="indeterminateDuration" format="integer" min="1" />
    Defines how the indeterminate mode should behave when the progress
    reaches max.
    <attr name="indeterminateBehavior">
        Progress starts over from 0.
        <enum name="repeat" value="1" />
        Progress keeps the current value and goes back to 0.
        <enum name="cycle" value="2" />
    </attr>
    <attr name="minWidth" format="dimension" />
    <attr name="maxWidth" />
    <attr name="minHeight" format="dimension" />
    <attr name="maxHeight" />
    <attr name="interpolator" format="reference" /> -->

```

Notice the use of the at-symbol (@) and the question-mark (?) to reference resources. The at-symbol indicates that we're referencing a resource previously defined else where (which may be from this project or from the Android framework). The question-mark indicates that we're referencing a resource value in the currently loaded theme. This is done by referring to a specific <item> by its name value. (E.g., panelTextColor uses the same color assigned to panelForegroundColor, defined before hand.) This technique can be used only in XML resources.

@引用资源（可能在系统框架、当前工程），? 表示在当前主题中引用资源，引用的是<item>标签的值，只能在xml文件中使用

```

动画图片是-animated-rotate xmlns:android="http://schemas.android.com/apk/res/android"
    android:drawable="@drawable/spinner_white_48"
    android:pivotX="50%"
    android:pivotY="50%"
    android:frameCount="12"
    android:frameDuration="100" />

```

注意最后两个图片不能用，类似的

```

android:textAppearance="?android:attr/textAppearanceLarge"
    android:gravity="center_vertical"
    android:paddingLeft="6dip"

```

theme.xml文件做了映射

使用:

```

android:textAppearance="?android:attr/textAppearanceLarge"
android:textAppearance="?android:attr/textAppearanceMedium"
android:textAppearance="?android:attr/textAppearanceSmall"
或
style="?android:attr/textAppearanceLarge"
style="?android:attr/textAppearanceMedium"
style="?android:attr/textAppearanceSmall"

```

appwidget-provider  
<http://dev.10086.cn/cmdn/wiki/index.php?doc-view-3645.html>  
<http://www.eoeandroid.com/thread-92760-1-1.html>  
[http://blog.sina.com.cn/s/blog\\_5688414b0100y127.html](http://blog.sina.com.cn/s/blog_5688414b0100y127.html)

```

-optimizationpasses 5
-dontusemixedcaseclassnames
-dontskipnonpubliclibraryclasses
-dontpreverify
-verbose
-optimizations !code/simplification/arithmetic,!field/*,!class/merging/*

```

```

-keep public class * extends android.app.Activity
-keep public class * extends android.app.Application
-keep public class * extends android.app.Service
-keep public class * extends android.content.BroadcastReceiver
-keep public class * extends android.content.ContentProvider
-keep public class * extends android.app.backup.BackupAgentHelper
-keep public class * extends android.preference.Preference
-keep public class org.apache.commons.codec.net.*
-keep public class org.apache.commons.codec.language.*
-keep public class org.apache.commons.codec.digest.*
-keep public class org.apache.commons.codec.binary.*
-keep public class org.apache.commons.codec.*
-keep public class org.apache.commons.httpclient.*
-keep public class org.apache.commons.httpclient.auth.*
-keep public class org.apache.commons.httpclient.cookie.*
-keep public class org.apache.commons.httpclient.methods.*
-keep public class org.apache.commons.httpclient.methods.multipart.*
-keep public class org.apache.commons.httpclient.params.*
-keep public class org.apache.commons.httpclient.protocol.*
-keep public class org.apache.commons.httpclient.util.*
-keep public class com.android.vending.licensing.LicensingService
-dontwarn java.awt.*
-keepclasseswithmembernames class * {
    native <methods>;
}

```

```

}

-keepclasseswithmembers class * {
    public <init>(android.content.Context, android.util.AttributeSet);
}

-keepclasseswithmembers class * {
    public <init>(android.content.Context, android.util.AttributeSet, int);
}

-keepclassmembers class * extends android.app.Activity {
    public void *(android.view.View);
}

-keepclassmembers enum * {
    public static **[] values();
    public static ** valueOf(java.lang.String);
}

-keep class * implements android.os.Parcelable {
    public static final android.os.Parcelable$Creator *;
}

```

在开发中，需要利用ContentProvider和其他应用通信。这样在执行query方法的时候，就会报错，cursor not closed。解决这个问题的办法是创建新的Cursor，返回给第三方；然后将本地Cursor close掉即可。代码如下：

```

String[] tableCursor = new String[] { "user_name", "user_pass", "update_time" };
MatrixCursor myNewCursor = new MatrixCursor(tableCursor);
if (cursor.moveToFirst()) {
    // 用户名
    String userName = cursor.getString(cursor.getColumnIndex("user_name"));
    // 密码
    String password = cursor.getString(cursor.getColumnIndex("user_pass"));
    String updateTime = cursor.getString(cursor.getColumnIndex("update_time"));
    myNewCursor.addRow(new Object[] { updateTime, userName, password });
}
cursor.close();
return myNewCursor;

```

an't find referenced class javax.servlet.ServletContextEvent  
[2012-03-26 21:23:33 - guoku] Warning: org.apache.commons.logging.impl.ServletContextCleaner: can't find referenced class javax.servlet.ServletContextEvent  
[2012-03-26 21:23:33 - guoku] Warning: org.apache.commons.logging.impl.LogKitLogger: can't find referenced class org.apache.log.Hierarchy  
[2012-03-26 21:23:33 - guoku] Warning: org.apache.commons.logging.impl.LogKitLogger: can't find referenced class org.apache.log.Hierarchy  
[2012-03-26 21:23:33 - guoku] Warning: org.apache.commons.logging.impl.LogKitLogger: can't find referenced class org.apache.log.Logger

<http://www.iteye.com/topic/1012395>  
<http://proguard.sourceforge.net/index.html#manual/examples.html>  
<http://www.cnblogs.com/over140/archive/2011/04/22/2024528.html#ConfiguringProguard>  
<http://developer.android.com/guide/developing/tools/proguard.html> google自己的

<!-- <http://www.cnblogs.com/qianxudetianxia/archive/2011/09/19/2068760.html>  
<http://www.cnblogs.com/loulijun/archive/2012/03/20/2320838.html>  
<http://hi.baidu.com/huaxinchang/blog/item/ea528e595b8963d2934f051.html>  
<http://hi.baidu.com/huaxinchang/blog/item/04a358fd55c0640108244d52.html>  
<http://www.programmer.com.cn/8852/8672/>  
<http://blog.lytsing.org/archives/559.html>  
<http://blog.csdn.net/ichliebphone/article/details/6591071> -->

<http://www.vogella.de/articles/AndroidCloudToDeviceMessaging/article.html> 例子  
[http://www.vogella.de/articles/AndroidCloudToDeviceMessaging/article.html#tutorial\\_serverapp](http://www.vogella.de/articles/AndroidCloudToDeviceMessaging/article.html#tutorial_serverapp)

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home/wiki/Android1/page/%E5%90%84%E7%A7%8D%E6%9D%83%E9%99%90?lang=en> 权限大全  
<http://www.vogella.de/android.html> 例子  
[http://bigcat.easymorse.com/?page\\_id=1477](http://bigcat.easymorse.com/?page_id=1477) 例子  
<http://bigcat.easymorse.googlecode.com/svn/trunk/>  
<http://stackoverflow.com/questions/7994959/how-to-add-section-header-to-custom-list-view-in-android/8047328#8047328>

<http://code.google.com/p/androidcontacts/source/browse/#svn%2Ftrunk%2FAndroidContacts%2F%7E7%BB%86%E5%8C%96%E8%BF%AD%E4%B8%A3%E6%96%87%E6%A1%A3> 联系人  
<http://code.google.com/p/Fxcontacts/source/browse/#svn%2Ftrunk%2Fsrc%2Fcom%2Fandroi.d%2Ffxcontacts%2Fmodel> 联系人  
<https://www.codeaurora.org/glt/projects/tlroy-gb-dsds-7227/repository/revisions/c0b022e5852458993d102e0dfe78d07ca1701ef2/show/packages/apps> 所有app源代码

<http://code.google.com/p/live/wiki/AccessingTheSource?tm=4> Five'

```

gb.appendWhere("param1=1");
gb.appendWhere(" AND param2=1");

```

```

etSearch.addTextChangedListener(new TextWatcher() {
    // 在s中，count个始于start的字符被旧字符串中长度为before的若干个字符。在这个回调中改变s会出错，s是最终的结果
    @Override
    public void onTextChanged(CharSequence s, int start, int before,
        int count) {
        // TODO Auto-generated method stub
        Log.e("TextWatcher", "CharSequence s:" + s + ", start:" + start
            + " before:" + before + " count:" + count);
    }

    // 在s，count个始于start的字符被长度为after的新字符串替代。It is an error to attempt
    // to make changes to s from this callback.
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count,
        int after) {
        // TODO Auto-generated method stub
        Log.e("TextWatcher", "CharSequence s:" + s + ", start:" + start
            + " after:" + after + " count:" + count);
    }

    // legitimate to make further changes to s from this callback, but
    // be careful not to get yourself into an infinite loop, because any
    // changes you make will cause this method to be called again
    // recursively.
    @Override
    public void afterTextChanged(Editable s) {
        // TODO Auto-generated method stub
        Log.e("TextWatcher", "CharSequence s:" + s);
        String searchContent = s.toString();
        showResults(sinaUser.getId(), searchContent);
    }
}

```

```

// 输入一个字符 before: 0 (count, 空格) 个字符被一个 (after) 字符替换; on: 一个 (count) 字符替换了 0 个 (before, 空格) 字符
// 删除一个字符 before: 1 (count) 个字符被 0 个 (after, 空格) 替换; on:
// 0 个 (count, 空格) 替换了 1 个 (before) 字符
// CharSequence s:, start:0 after:1 count:0
// CharSequence s:a, start:0 before:0 count:1
// CharSequence s:a
// CharSequence s:a, start:1 after:1 count:0
// CharSequence s:ab, start:1 before:0 count:1
// CharSequence s:ab
// CharSequence s:ab, start:1 after:0 count:1
// CharSequence s:a, start:1 before:1 count:0
// CharSequence s:a
// CharSequence s:a, start:0 after:0 count:
// CharSequence s:, start:0 before:1 count:0

});

// Uri uri = ContentUris.withAppendedId(FriendsProvider.CONTENT_URI, id);
// Uri uri2 = Uri.withAppendedPath(FriendsProvider.CONTENT_URI, Uri.encode(id+"?" + FriendsProvider.LIMIT+"=3,50"));

```

文字高亮

```

1. String highLightStr = "HighLight MaYingCai";
2. String highLight = "MaYingCai";
3.
4. int start = highLightStr.indexOf(highLight);
5.
6. SpannableStringBuilder style = new SpannableStringBuilder(highLightStr);
7. style.setSpan(new BackgroundColorSpan(Color.RED), start, start + highLight.length(), Spannable.SPAN_EXCLUSIVE_EXCLUSIVE);
8.
9. highLightView.setText(style);

```

<http://commonsware.com/AdVAndroid/> 创建重用的组件

TextView属性:

`android:drawSelectorOnTop="false" ?`

```

// content://com.example.android.searchabledict.DictionaryProvider/search_suggest_query?limit=50
// String auth = uri.getAuthority();
// String fragment = uri.getFragment();
// String host = uri.getHost();
// String lastPathSegment = uri.getLastPathSegment();
// String query2 = uri.getQuery();
// String schemeSpecificPart = uri.getSchemeSpecificPart();
// String path = uri.getPath();
// String scheme = uri.getScheme();
// String userInfo = uri.getUserInfo();
// int port = uri.getPort();
//
// Log.e("scheme", scheme);//content
// Log.e("schemeSpecificPart", schemeSpecificPart);// //com.example.android.searchabledict.DictionaryProvider/search_suggest_query?limit=50
// Log.e("host", host);//com.example.android.searchabledict.DictionaryProvider
// Log.e("auth", auth);//com.example.android.searchabledict.DictionaryProvider
// Log.e("port", port+"");//-1
// Log.e("fragment", "fragment:"+fragment);// 空串
// Log.e("path", path);// /search_suggest_query注意有/
// Log.e("lastPathSegment", lastPathSegment);//search_suggest_query 注意没有/
// Log.e("query", query);//limit=50
// Log.e("userInfo", "+userInfo");

```

## android: adding button to the title of the app?

<http://stackoverflow.com/questions/2569753/android-adding-button-to-the-title-of-the-app>

**Try to use Window.FEATURE\_CUSTOM\_TITLE but got Exception:You cannot combine custom titles with other title feature..**

<http://stackoverflow.com/questions/2686556/try-to-use-window-feature-custom-title-but-got-exceptionyou-cannot-combine-cust>

**ACTION\_IMAGE\_CAPTURE with EXTRA\_OUTPUT still generating jpg in Camera's default folders**

<http://code.google.com/p/android/issues/detail?id=19268>

**Camera preview does not work on HTC Desire**

<http://code.google.com/p/android/issues/detail?id=7909>

**Camera crashes in Froyo emulator**

<http://code.google.com/p/android/issues/detail?id=9376>

## How to set icon to title bar for each Activity in TabLayout

<http://stackoverflow.com/questions/4533830/how-to-set-icon-to-title-bar-for-each-activity-in-tablayout>

<http://www.edumobile.org/android/android-programming-tutorials/creating-a-custom-title-bar/>

```

//http://walletapp.net/en/cookbook/crop-image(https://github.com/biokys/cropimage)
//http://code.google.com/p/mobileocr/source/checkout
//http://code.google.com/p/wwwjdic/source/checkout
//https://android-image-editor.googlecode.com/svn/trunk/

//https://github.com/eleybourn/Book-Catalogue
//https://github.com/biokys/cropimage
//https://github.com/lorensiuswlt/AndroidImageCrop(博客http://www.londatiga.net/featured-articles/how-to-select-and-crop-image-on-android/. http://www.londatiga.net/it/how-to-create-android-image-picker/)

```

`new GradientDrawable().setCornerRadius(radius)`

GradientDrawable 支持使用渐变色来绘制图形，通常可以用作Button或是背景图形。

GradientDrawable的构造方法:

```

public GradientDrawable() {
    this(new GradientState(Orientation.TOP_BOTTOM, null)); //默认从顶部到底部
}

```

```

二、
public GradientDrawable(Orientation orientation, int[] colors) {
    this(new GradientState(orientation, colors));
}
}
调用下面:
private GradientDrawable(GradientState state) {
    mGradientState = state;
    initializeWithState(state);
    mRectIsDirty = true;
}
}

```

```

GradientDrawable的内部类:
final static class GradientState extends ConstantState {

    public int mShape = RECTANGLE; //默认形状为矩形
    public int mGradient = LINEAR_GRADIENT; //默认是线性渐变
    private float mCenterX = 0.5f; //默认中心
    private float mCenterY = 0.5f;
    private float mGradientRadius = 0.5f; //默认渐变弧度0.5

    GradientState() {
        mOrientation = Orientation.TOP_BOTTOM;
    }

    GradientState(Orientation orientation, int[] colors) {
        mOrientation = orientation;
        mColors = colors;
    }

    public GradientState(GradientState state) {
        mChangingConfigurations = state.mChangingConfigurations;
        mShape = state.mShape;
        mGradient = state.mGradient;
        mOrientation = state.mOrientation;
        if (state.mColors != null) {
            mColors = state.mColors.clone();
        }
        if (state.mPositions != null) {
            mPositions = state.mPositions.clone();
        }
        mHasSolidColor = state.mHasSolidColor;
        mStrokeWidth = state.mStrokeWidth;
        mStrokeColor = state.mStrokeColor;
        mStrokeDashWidth = state.mStrokeDashWidth;
        mStrokeDashGap = state.mStrokeDashGap;
        mRadius = state.mRadius;
        if (state.mRadiusArray != null) {
            mRadiusArray = state.mRadiusArray.clone();
        }
        if (state.mPadding != null) {
            mPadding = new Rect(state.mPadding);
        }
        mWidth = state.mWidth;
        mHeight = state.mHeight;
        mInnerRadiusRatio = state.mInnerRadiusRatio;
        mThicknessRatio = state.mThicknessRatio;
        mInnerRadius = state.mInnerRadius;
        mThickness = state.mThickness;
        mCenterX = state.mCenterX;
        mCenterY = state.mCenterY;
        mGradientRadius = state.mGradientRadius;
        mUseLevel = state.mUseLevel;
        mUseLevelForShape = state.mUseLevelForShape;
    }
}

```

所以new GradientDrawable()创建一个矩形, 线性渐变色, 改变图形:  
new GradientDrawable().setShape(GradientDrawable.LINE); GradientDrawable允许指定绘制图形的种类: LINE, OVAL, RECTANGLE或是RING, 颜色渐变支持LINEAR\_GRADIENT, RADIAL\_GRADIENT 和 SWEEP\_GRADIENT 三种渐变模式, 分别表示线性, 圆形, 弧形 (扫过去)。  
其中在使用RECTANGLE (矩形), 还允许设置矩形四个角为圆角, 每个圆角的半径可以分别设置: public void setCornerRadii(float[] radii), 共8个值, 每个点有x和y。四个圆角的顺序为左上, 右上, 右下, 左下。如果X\_Radius, Y\_Radius为0表示还是直角。

还指定渐变方向: TOP\_BOTTOM, TR\_BR, RIGHT\_LEFT, BR\_TL, BOTTOM\_TOP, BL\_TR, LEFT\_RIGHT, TL\_BR,

public GradientDrawable(GradientDrawable.Orientation orientation, int[] colors), orientation指定了渐变的方向 (以上八种), 渐变的颜色由colors数组指定, 数组中的每个值为一个颜色。

```

定义一个渐变方向从组左上到右下, 渐变颜色为红, 绿, 蓝三色:
mDrawable = new GradientDrawable(GradientDrawable.Orientation.TL_BR,
    new int[] { 0xFF0000, 0x00FF00, 0x0000FF });

mDrawable.setGradientType(GradientDrawable.LINEAR_GRADIENT);
setCornerRadii(mDrawable, r, r, 0, 0);
mDrawable.draw(canvas);
...
mDrawable.setGradientType(GradientDrawable.RADIAL_GRADIENT);
setCornerRadii(mDrawable, 0, 0, r, r);
mDrawable.draw(canvas);
...
mDrawable.setGradientType(GradientDrawable.SWEEP_GRADIENT);
setCornerRadii(mDrawable, 0, r, r, 0);
mDrawable.draw(canvas);
...
mDrawable.setGradientType(GradientDrawable.LINEAR_GRADIENT);
setCornerRadii(mDrawable, r, 0, 0, r);
mDrawable.draw(canvas);
...
mDrawable.setGradientType(GradientDrawable.RADIAL_GRADIENT);
setCornerRadii(mDrawable, r, 0, r, 0);
mDrawable.draw(canvas);
...
mDrawable.setGradientType(GradientDrawable.SWEEP_GRADIENT);
setCornerRadii(mDrawable, 0, r, 0, r);
mDrawable.draw(canvas);
...
static void setCornerRadii(GradientDrawable drawable,
    float r0, float r1, float r2, float r3) {
    drawable.setCornerRadii(new float[] { r0, r0, r1, r1,
        r2, r2, r3, r3 });
}
http://www.imobilebbs.com/wordpress/?p=1620

```

```

./dzj-dex2jar.sh hw-org.dayup.gnotes.dex
./dzj-dex2jar.sh hw-my.handrite.dex

```

```

./daj-dexzjar.sh hw-mobi.qiss.qmemo.dex
./daj-dexzjar.sh hw-com.zenpie.genialwriting.dex
./daj-dexzjar.sh hw-com.evernote.world.dex
./daj-dexzjar.sh hw-com.dansi.Drawer.dex
./daj-dexzjar.sh hw-com.bejoy.myletter.dex

./dexzjar.sh hw-com.bejoy.myletter.dex

```

上传文件

```

import org.apache.commons.httpclient.Header;
import org.apache.commons.httpclient.methods.PostMethod;
import org.apache.commons.httpclient.methods.multipart.FilePart;
import org.apache.commons.httpclient.methods.multipart.MultipartRequestEntity;
import org.apache.commons.httpclient.methods.multipart.Part;
import org.apache.commons.httpclient.methods.multipart.PartBase;
import org.apache.commons.httpclient.methods.multipart.StringPart;

// PostParameter, ImageItem以及ByteArrayPart, Response都是自定义的
public Response multipartURL(String url, PostParameter[] params, ImageItem item, boolean authenticated) throws WeiboException{
    PostMethod post = new PostMethod(url);
    try {
        org.apache.commons.httpclient.HttpClient client = new org.apache.commons.httpclient.HttpClient();
        long t = System.currentTimeMillis();
        Part[] parts=null;
        if(params==null){
            parts=new Part[1];
        }else{
            parts=new Part[params.length+1];
        }
        if (params != null ) {
            int i=0;
            for (PostParameter entry : params) {
                parts[i++]=new StringPart( entry.getName(),(String)entry.getValue());
            }
            parts[parts.length-1]=new ByteArrayPart(item.getContent(), item.getName(), item.getImageType());
        }
        post.setRequestEntity( new MultipartRequestEntity(parts, post.getParams()) );
        List<Header> headers = new ArrayList<Header>();

        if (authenticated) {
            if (oauth == null) {
            }
            String authorization = null;
            if (null != oauth) {
                // use OAuth
                authorization = oauth.generateAuthorizationHeader( "POST" , url, params, oauthToken);
            } else {
                throw new IllegalStateException(
                    "Neither user ID/password combination nor OAuth consumer key/secret combination supplied");
            }
            headers.add(new Header("Authorization", authorization));
            log("Authorization: " + authorization);
        }
        client.getHostConfiguration().getParams().setParameter("http.default-headers", headers);
        client.executeMethod(post);

        Response response=new Response();
        response.setResponseAsString(post.getResponseAsString());
        response.setStatusCode(post.getStatusCode());

        log("multipartURL URL:" + url + ", result:" + response + ", time:" + (System.currentTimeMillis() - t));
        return response;
    } catch (Exception ex) {
        throw new WeiboException(ex.getMessage(), ex, -1);
    } finally {
        post.releaseConnection();
    }
}

public Response multipartURL(String fileName,String url, PostParameter[] params,File file,boolean authenticated) throws WeiboException{
    PostMethod post = new PostMethod(url);
    org.apache.commons.httpclient.HttpClient client = new org.apache.commons.httpclient.HttpClient();
    try {
        long t = System.currentTimeMillis();
        Part[] parts=null;
        if(params==null){
            parts=new Part[1];
        }else{
            parts=new Part[params.length+1];
        }
        if (params != null ) {
            int i=0;
            for (PostParameter entry : params) {
                parts[i++]=new StringPart( entry.getName(),(String)entry.getValue());
            }
        }
        FilePart filePart=new FilePart(fileName,file.getName(), file,new FileType().getMIMEType(file),"UTF-8");
        filePart.setTransferEncoding("binary");
        parts[parts.length-1]= filePart;

        post.setRequestEntity( new MultipartRequestEntity(parts, post.getParams()) );
        List<Header> headers = new ArrayList<Header>();

        if (authenticated) {
            if (oauth == null) {
            }
            String authorization = null;
            if (null != oauth) {
                // use OAuth
                authorization = oauth.generateAuthorizationHeader( "POST" , url, params, oauthToken);
            } else {
                throw new IllegalStateException(
                    "Neither user ID/password combination nor OAuth consumer key/secret combination supplied");
            }
            headers.add(new Header("Authorization", authorization));
            log("Authorization: " + authorization);
        }
        client.getHostConfiguration().getParams().setParameter("http.default-headers", headers);
        client.executeMethod(post);

        Response response=new Response();
        response.setResponseAsString(post.getResponseAsString());
        response.setStatusCode(post.getStatusCode());
    }
}

```

```

        log("multipartURL URL:" + url + ", result:" + response + ", time:" + (System.currentTimeMillis() - t));
        return response;
    } catch (Exception ex) {
        throw new WeiboException(ex.getMessage(), ex, -1);
    } finally {
        post.releaseConnection();
        client=null;
    }
}

System.setProperty("sun.net.client.defaultConnectTimeout", "5000");
System.setProperty("sun.net.client.defaultReadTimeout", "2000");

HttpClient client = new DefaultHttpClient();
client.getParams().setParameter(CoreConnectionPNames.CONNECTION_TIMEOUT, NetConfig.CONNECTION_TIMEOUT); //"http.connection.timeout"
client.getParams().setParameter(CoreConnectionPNames.SO_TIMEOUT, NetConfig.SO_TIMEOUT); //"http.socket.timeout"

conn.setInstanceFollowRedirects(true);

```

```

public class ApplicationEx extends Application {
    private HttpClient httpClient ;
    @Override
    public void onCreate() {
        super.onCreate();

        createClient();
    }

    @Override
    public void onLowMemory() {
        super.onLowMemory();
        shutdownHttpClient();
    }

    @Override
    public void onTerminate() {
        super.onTerminate();
        shutdownHttpClient();
    }

    private void createClient(){
        HttpParams params = new BasicHttpParams();
        ConnManagerParams.setMaxTotalConnections(params, 100);
        HttpProtocolParams.setVersion(params, HttpVersion.HTTP_1_1);
        HttpProtocolParams.setContentCharset (params, HTTP.UTF_8); //plus
        HttpProtocolParams.setUseExpectContinue (params, true); //plus

        SchemeRegistry schemeRegistry = new SchemeRegistry();
        schemeRegistry.register(new Scheme("http", PlainSocketFactory.getSocketFactory(), 80));
        schemeRegistry.register(new Scheme("https", SSLSocketFactory.getSocketFactory(), 443));

        ClientConnectionManager cm = new ThreadSafeClientConnManager(params, schemeRegistry);
        this.httpClient = new DefaultHttpClient(cm, params);
    }

    private void shutdownHttpClient(){
        ClientConnectionManager connectionManager = httpClient.getConnectionManager();
        connectionManager.shutdown();
    }
}

```

```

    圓角
    protected void onDraw(Canvas canvas) {
        Path clipPath = new Path();
        int w = this.getWidth();
        int h = this.getHeight();
        clipPath.addRoundRect(new RectF(0,0,w,h), 10.0f, 10.0f, Path.Direction.CW);
        canvas.clipPath(clipPath);
        super.onDraw(canvas);
    }

```

For a more controlled method draw a rounded rectangle and mask it onto your image using the porter-duff Xfer mode of the paint.

```

First setup the Xfer paint and the rounded bitmap:
Bitmap myCoolBitmap = ... ; // <-- Your bitmap you want rounded
int w = myCoolBitmap.getWidth(), h = myCoolBitmap.getHeight();

// We have to make sure our rounded corners have an alpha channel in most cases
Bitmap rounder = Bitmap.createBitmap(w,h,Bitmap.Config.ARGB_8888);
Canvas canvas = new Canvas(rounder);

// We're going to apply this paint eventually using a porter-duff xfer mode.
// This will allow us to only overwrite certain pixels. RED is arbitrary. This
// could be any color that was fully opaque (alpha = 255)
Paint xferPaint = new Paint(Paint.ANTI_ALIAS_FLAG);
xferPaint.setColor(Color.RED);

// We're just reusing xferPaint to paint a normal looking rounded box, the 20.f
// is the amount we're rounding by.
canvas.drawRoundRect(new RectF(0,0,w,h), 20.0f, 20.0f, xferPaint);

// Now we apply the 'magic sauce' to the paint
xferPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.DST_IN));
Now apply this bitmap ontop of your image:

```

```

canvas.drawBitmap(myCoolBitmap, 0,0, null);
canvas.drawBitmap(rounder, 0, 0, xferPaint);
Note: this is not the same canvas as before, it is a canvas to whatever buffer you're drawing to.

```

```

public static Bitmap getRoundedCornerBitmap(Bitmap bitmap) {
    Bitmap output = Bitmap.createBitmap(bitmap.getWidth(),
        bitmap.getHeight(), Config.ARGB_8888);
    Canvas canvas = new Canvas(output);

    final int color = 0xff424242;
    final Paint paint = new Paint();
    final Rect rect = new Rect(0, 0, bitmap.getWidth(), bitmap.getHeight());
    final RectF rectF = new RectF(rect);
    final float roundPx = 12;

    paint.setAntiAlias(true);

```



```

        canvas.drawARGB(0, 0, 0, 0);
        paint.setColor(color);

        canvas.drawRoundRect(rectF, roundPx, roundPx, paint);

        paint.setXfermode(new PorterDuffXfermode(Mode.SRC_IN));
        canvas.drawBitmap(bitmap, rect, rect, paint);

        return output;
    }
}

```

<http://stackoverflow.com/questions/1705239/how-should-i-give-images-rounded-corners-in-android>

```

public class StartActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new TouchView(this));
    }

    class TouchView extends View{
        Bitmap bgr;
        Bitmap overlayDefault;
        Bitmap overlay;
        Paint pTouch;
        int X = -100;
        int Y = -100;
        Canvas c2;

        public TouchView(Context context) {
            super(context);

            bgr = BitmapFactory.decodeResource(getResources(), R.drawable.bgr);
            overlayDefault = BitmapFactory.decodeResource(getResources(), R.drawable.overlay);
            overlay = BitmapFactory.decodeResource(getResources(), R.drawable.overlay).copy(Config.ARGB_8888, true);
            c2 = new Canvas(overlay);

            pTouch = new Paint(Paint.ANTI_ALIAS_FLAG);
            pTouch.setXfermode(new PorterDuffXfermode(Mode.SRC_OUT));
            pTouch.setColor(Color.TRANSPARENT);
            pTouch.setMaskFilter(new BlurMaskFilter(15, Blur.NORMAL));

        }

        @Override
        public boolean onTouchEvent(MotionEvent ev) {

            switch (ev.getAction()) {

                case MotionEvent.ACTION_DOWN: {

                    X = (int) ev.getX();
                    Y = (int) ev.getY();
                    invalidate();

                    break;

                }

                case MotionEvent.ACTION_MOVE: {

                    X = (int) ev.getX();
                    Y = (int) ev.getY();
                    invalidate();
                    break;

                }

                case MotionEvent.ACTION_UP:

                    break;

            }

            return true;
        }

        @Override
        public void onDraw(Canvas canvas){
            super.onDraw(canvas);

            //draw background
            canvas.drawBitmap(bgr, 0, 0, null);
            //copy the default overlay into temporary overlay and punch a hole in it
            c2.drawBitmap(overlayDefault, 0, 0, null); //exclude this line to show all as you draw
            c2.drawCircle(X, Y, 80, pTouch);
            //draw the overlay over the background
            canvas.drawBitmap(overlay, 0, 0, null);

        }

    }

}

```

[android.setTag\(int key, Object tag\)抛出IllegalArgumentException的根源](#)

setTag是android的view类中很有用的一个方法，可以用它来给空间附加一些信息，在很多场合下都得到妙用。

setTag (Object tag)方法比较简单，这里主要谈一谈带两个参数的setTag方法。

官方的api文档中提到：“The specified key should be an id declared in the resources of the application to ensure it is unique (see the [ID resource type](#)). Keys identified as belonging to the Android framework or not associated with any package will cause an [IllegalArgumentException](#) to be thrown. 所以抛出IllegalArgumentException的原因就在于key不唯一，那么如何保证这种唯一性呢？定义一个final类型的int变量和硬编码一个值的方式都是行不通的。

```

private static final int TAG_ONLINE_ID = 1;
((Button) row.findViewById(R.id.btnPickContact)).setTag(TAG_ONLINE_ID,objContact.onlineid);
05-18 20:29:38.044: ERROR/AndroidRuntime(5453): java.lang.IllegalArgumentException: The key must be an application-specific resource id.
05-18 20:29:38.044: ERROR/AndroidRuntime(5453): at android.view.View.setTag(View.java:7704)
05-18 20:29:38.044: ERROR/AndroidRuntime(5453): at com.mypkg.viewP.inflateRow(ViewP.java:518)

```

如果只需要设置一个tag，那么直接调用setTag (Object tag)方法就可以轻松搞定，如果一定需要使用多个tag绑定，那么需要先在res/values/strings.xml中添加

```

<resources>
<item type="id" name="tag_first"></item>
<item type="id" name="tag_second"></item>
</resources>
使用的时候写成

```

```

imageView.setTag(R.id.tag_first, "Hello");
imageView.setTag(R.id.tag_second, "Success");

```

就可以了

shape的bug

While this question has been answered already (it's a bug that causes bottomLeftRadius and bottomRightRadius to be reversed), the bug has been fixed in android 3.1 (api level 12 - tested on the emulator).

So to make sure your drawables look correct on all platforms, you should put "corrected" versions of the drawables (i.e. where bottom left/right radii are actually correct in the xml) in the res/drawable-v12 folder of your app. This way all devices using an android version >= 12 will use the correct drawable files, while devices using older versions of android will use the "workaround" drawables that are located in the res/drawables folder.

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >

    <corners
        android:bottomLeftRadius="0.0dip"
        android:bottomRightRadius="10.0dip"
        android:radius="1.0dip"
        android:topLeftRadius="0.0dip"
        android:topRightRadius="0.0dip"
    />

    <solid android:color="@color/light_grey_background" />

</shape>
android 3.1 之前bottomLeftRadius和bottomRightRadius是相反的，3.1之后是正确的，所以需要建立两个文件夹res/drawable-v12和res/drawable。
drawable-v11
<?xml version="1.0" encoding="utf-8"?>
<shape android:shape="rectangle"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <corners android:radius="1.0dip" android:topLeftRadius="0.0dip" android:topRightRadius="0.0dip" android:bottomLeftRadius="10.0dip" android:bottomRightRadius="0.0dip"/>
    <solid android:color="@color/light_grey_background" />
</shape>
drawable
<?xml version="1.0" encoding="utf-8"?>
<shape android:shape="rectangle"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <corners android:radius="1.0dip" android:topLeftRadius="0.0dip" android:topRightRadius="0.0dip" android:bottomLeftRadius="0.0dip" android:bottomRightRadius="10.0dip" />
    <solid android:color="@color/light_grey_background" />
</shape>

<?xml version="1.0" encoding="utf-8"?>
<resources>
<item type="drawable" name="white">#ffffff</item>
<item type="drawable" name="black">#ff000000</item>
<item type="drawable" name="gray">#ffcccc</item>
<item type="drawable" name="bg_white">#ffffff</item>
<item type="drawable" name="view_border">#ffff0000</item>
<item type="drawable" name="transparent">#00000000</item>
<item type="drawable" name="section_select_title_bottomline">#ff8c8c</item>
<item type="drawable" name="register_view_background">#ffb8b8</item>
<item type="drawable" name="title_icon_pressed">#ff094271</item>
<item type="drawable" name="sectiongroup_bg_white">#ffffff</item>
<item type="drawable" name="list_selector_background">#ffcedeb</item>
<item type="drawable" name="setting_bg_listview">#ffffff</item>
<item type="drawable" name="setting_bg_layout">#ffcfcd</item>
<item type="drawable" name="night_bg_white">#ff1d1d</item>
<item type="drawable" name="night_view_border">#ff9595</item>
<item type="drawable" name="night_title_icon_pressed">#ff1d1d</item>
<item type="drawable" name="night_global_divider">#ff3131</item>
<item type="drawable" name="night_sectiongroup_bg_white">#ffc6c6</item>
<item type="drawable" name="night_sectiongroup_global_divider">#ffacac</item>
<item type="drawable" name="night_list_selector_background">#ff9494</item>
<item type="drawable" name="night_setting_bg_listview">#ff0000</item>
<item type="drawable" name="night_setting_bg_layout">#ff1e1e</item>
</resources>

```

自定义dialog

```

dialog = new Dialog(FloatActivity.this,R.style.ProgressDialog);
<style name="ProgressDialog">
    <item name="android:windowFrame">@null</item>
    <item name="android:windowBackground">@android:color/transparent</item>
    <item name="android:windowIsFloating">true</item>
    <item name="android:windowContentOverlay">@null</item>
    <item name="android:windowTitleStyle">@null</item>
    <item name="android:windowAnimationStyle">@android:style/Animation.Dialog</item>
    <item name="android:windowSoftInputMode">stateUnspecified|adjustPan</item>
    <item name="android:backgroundDimEnabled">false</item>
    <item name="android:background">@android:color/transparent</item>
</style>

<style name="Theme_TranslucentDlg" parent="@android:style/Theme.Dialog">
    <item name="android:windowBackground">@android:color/transparent</item>
    <item name="android:windowFrame">@null</item>
    <item name="android:windowNoTitle">true</item>
    <item name="android:windowIsFloating">true</item>
    <item name="android:windowIsTranslucent">true</item>
    <item name="android:windowContentOverlay">@null</item>
    <item name="android:windowAnimationStyle">@android:style/Animation.Dialog</item>
    <item name="android:backgroundDimEnabled">true</item>
    <item name="android:windowSoftInputMode">stateUnchanged|adjustResize</item>
</style>

```

```

package com.sunxu.org.IndividualityDialog;

```

```

import android.app.Activity;
import android.app.Dialog;
import android.content.Context;
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;

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

        Button btn = (Button)findViewById(R.id.button1);

        btn.setOnClickListener(new OnClickListener()
        {
            public void onClick(View v)
            {

```

```

        //多个Activity嵌套时用this.parent否则异常
        new myDialog(IndividualityDialogActivity.this)
            .showDialog(R.layout.dialog, 80, 50);
    }
}

//自定义Dialog
class myDialog extends Dialog{

    private Window window = null;

    public myDialog(Context context)
    {
        super(context);
    }

    public void showDialog(int layoutResID, int x, int y){
        setContentView(layoutResID);

        windowDeploy(x, y);

        //设置触摸对话框意外的地方取消对话框
        setCanceledOnTouchOutside(true);
        show();
    }

    //设置窗口显示
    public void windowDeploy(int x, int y){
        window = getWindow(); //得到对话框
        window.setWindowAnimations(R.style.dialogWindowAnim); //设置窗口弹出动画
        window.setBackgroundDrawableResource(R.color.vifrifcation); //设置对话框背景为透明
        WindowManager.LayoutParams wl = window.getAttributes();
        //根据x, y坐标设置窗口需要显示的位置
        wl.x = x; //x小于0左移, 大于0右移
        wl.y = y; //y小于0上移, 大于0下移
        wl.alpha = 0.6f; //设置透明度
        wl.gravity = Gravity.BOTTOM; //设置重力
        window.setAttributes(wl);
    }
}

}

<resources>
<style name="dialogWindowAnim" parent="android:Animation" mce-bogus="1">
<item name="android:windowEnterAnimation">@anim/dialog_enter_anim</item>
<item name="android:windowExitAnimation">@anim/dialog_exit_anim</item>
</style>
</resources>

<set xmlns:android="http://schemas.android.com/apk/res/android">
<scale
    android:interpolator="@android:anim/accelerate_interpolator"
    android:fromXScale="1.0"
    android:toXScale="1.0"
    android:fromYScale="0.0"
    android:toYScale="1.0"
    android:pivotX="0%"
    android:pivotY="100%"
    android:fillAfter="false"
    android:duration="400"/>
</set>
<set xmlns:android="http://schemas.android.com/apk/res/android">
<scale
    android:interpolator="@android:anim/accelerate_interpolator"
    android:fromXScale="1.0"
    android:toXScale="1.0"
    android:fromYScale="1.0"
    android:toYScale="0.0"
    android:pivotX="0%"
    android:pivotY="100%"
    android:fillAfter="false"
    android:duration="400"/>
</set>
<?xml version="1.0" encoding="utf-8"?>
<resources>
<color name="vifrifcation">#00000000</color> <!-- 透明 -->
</resources>

```

<http://androiddev.orkitra.com/?p=111>

## Avoid dialog leak issue

Dialog leak issue usually occur when the dialog is shown and later the activity destroy without dismissing it. Most common leaks happen while dialog is showing then the configuration or orientation change occur. So, it causes the leak and sometime it may also cause the App to crash or go into an invalid state. You may notice the leak only by looking at the logcat. This mean that you do not dismiss your dialog correctly before your activity get destroy.

When the orientation change, the activity will be destroyed and re-created again. Before the activity is destroyed due to orientation change, it will save the current state first. So, when the activity is re-created it tries to restore instance state (onRestoreInstanceState()) from the previous activity. Notice that when the activity is re-created, it is a new activity and a new view. The exception may occur, if it try to update the old view (possibly, the dialog that did not dismiss) in the new activity, because the old view will have an invalid Context.

So, if your application have any pop up dialog, make sure that you dismiss it before your application get destroy. Here is what you can do.

1. Dismiss all dialogs completely if exist in onPause() or immediately after it is no longer need.
2. Since most common leaks happen while dialog is showing then the configuration or orientation change occur, another way to avoid dialog leak is NOT to destroy the activity when orientation change. If the activity is not destroyed, then there is no dialog leak and there is no re-creation of new activity, thus no issue neither. This can be achieved by using the attribute android:configChanges="orientation|keyboardHidden" in the manifest file and using the Override method onConfigurationChanged() inside your activity. By using the attribute android:configChanges you will tell Android that you will handle the configuration change yourself so, don't destroy the activity. And you can handle the configuration change in the Override method, or you may leave it blank if you prefer not to do anything.

## Handle configuration or orientation change

When configuration or orientation change happen, the activity will be automatically destroyed and re-created again. This behavior may fit to some activity but not all. Sometimes, you want your activity to handle configuration or orientation change instead of letting android automatically handle it. This way your activity will have freedom to do what you want. Below is an example of how to handle configuration or orientation change:

First, you need to modify AndroidManifest.xml by putting a line of code below inside your activity property. By using this "configChanges" attribute in your activity, you will tell Android that do not do anything when orientation change or keyboardHidden, so your activity will not get destroy when switching orientation. Instead you need to handle the orientation change yourself.

```
android:configChanges="orientation|keyboardHidden"
```

Now, you are ready to handle any configuration change yourself by overriding method onConfigurationChanged() in your activity .java file. See example below:

```

public class MyActivity extends Activity {
    ...

    @Override
    public void onConfigurationChanged(Configuration newConfig) {
        // do something or nothing when configuration change.
        super.onConfigurationChanged(newConfig);
    }
}

```

```

public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);

    // Checks the orientation of the screen
    if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE) {
        Toast.makeText(this, "landscape", Toast.LENGTH_SHORT).show();
    } else if (newConfig.orientation == Configuration.ORIENTATION_PORTRAIT){
        Toast.makeText(this, "portrait", Toast.LENGTH_SHORT).show();
    }
}

```

Notice that if you only have the android:configChanges="orientation|keyboardHidden" attribute in your XML file and not the Override method above (or having the Override method above but do nothing in there), Android will ignore any configuration change. That mean if you are in portrait mode and then switch to landscape mode, it will use portrait layout in landscape mode. And your activity will not get destroy. If you want to perform any action when orientation change, you may specify your action in the onConfigurationChanged() method.

Further, if you want to use onConfigurationChanged() method, you must put the attribute android:configChanges="orientation|keyboardHidden" in your XML file. Otherwise, the onConfigurationChanged() method will not get call at all.

QQ:

```

<?xml version="1.0" encoding="utf-8"?>
<rotate android:drawable="@drawable/waiting" android:useLevel="true" android:fromDegrees="0.0" android:toDegrees="360.0" android:pivotX="50.0%" android:pivotY="50.0%"
xmlns:android="http://schemas.android.com/apk/res/android" />

```