

Android —— TextView

一、TextView的API

1.1 结构

```
java.lang.Object
└─ android.view.View
   └─ android.widget.TextView
```

直接子类:

Button, CheckedTextView, Chronometer, DigitalClock, [EditText](#)


间接子类:

AutoCompleteTextView, CheckBox, CompoundButton, ExtractEditText, MultiAutoCompleteTextView, RadioButton, ToggleButton

1.2 API

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.+*/%\\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。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EditText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。EditText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 <pre> android:text="test_singleLine " android:singleLine="true" </pre> 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

<code>android:text</code>	设置显示文本.
<code>android:textAppearance</code>	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
<code>android:textColor</code>	设置文本颜色
<code>android:textColorHighlight</code>	被选中文字的底色，默认为蓝色
<code>android:textColorHint</code>	设置提示信息文字的颜色，默认为灰色。与hint一起使用。
<code>android:textColorLink</code>	文字链接的颜色.
<code>android:textScaleX</code>	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： <pre> abcde0.5f abcdef 1.0f abcdef 1.5f abcdef 2.0f </pre>
<code>android:textSize</code>	设置文字大小，推荐度量单位”sp”，如”15sp”
<code>android:textStyle</code>	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
<code>android:typeface</code>	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]
<code>android:height</code>	设置文本区域的高度，支持度量单位：px(像素)/dp/sp/in/mm(毫米)
<code>android:maxHeight</code>	设置文本区域的最大高度
<code>android:minHeight</code>	设置文本区域的最小高度
<code>android:width</code>	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 。
<code>android:maxWidth</code>	设置文本区域的最大宽度
<code>android:minWidth</code>	设置文本区域的最小宽度

1.3 补充说明

以下几个属性以及输入法相关的在这里都没有效果，在EditText将补充说明。`android:numeric`/`android:digits`/`android:phoneNumber`/`android:inputMethod`/`android:capitalize`/`android:autoText`

二、例子

2.1 跑马灯的效果

```
<TextView android:layout_width="100px"
    android:layout_height="wrap_content"
    android:textColor="@android:color/white"
    android:ellipsize="marquee"
    android:focusable="true"
    android:marqueeRepeatLimit="marquee_forever"
    android:focusableInTouchMode="true"
    android:scrollHorizontally="true"
    android:text="这才是真正的文字跑马灯效果"
>
</TextView>
```

[Android 2.2 API 中文文档系列 \(2\) —— EditText](#)

一、结构

```
java.lang.Object
└─ android.view.View
    └─ android.widget.TextView
        └─ android.widget.EditText
```

已知直接子类:

`AutoCompleteTextView`, `ExtractEditText`

已知间接子类:

`MultiAutoCompleteTextView`

二、xml属性

属性名称	描述
<code>android:autoLink</code>	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(<code>none/web/email/phone/map/all</code>)。这里只有在同时设置text时才自动识别链接，后来输入的无法自动识别。
<code>android:autoText</code>	自动拼写帮助。这里单独设置是没有效果的，可能需要其他输入法辅助才行，效果参见 视频 。

<p>android:bufferType</p>	<p>指定getText()方式取得的文本类别。选项 <code>editable</code> 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。<code>spannable</code> 则可在给定的字符区域使用样式，参见这里1、这里2。</p>
<p>android:capitalize</p>	<p>设置英文字母大写类型。设置如下值： <code>sentences</code>仅第一个字母大写；<code>words</code>每一个单词首字母大小，用空格区分单词；<code>characters</code>每一个英文字母都大写。在模拟器上用PC键盘直接输入可以出效果，但是用软键盘无效果。</p>
<p>android:cursorVisible</p>	<p>设定光标为显示/隐藏，默认显示。如果设置 <code>false</code>，即使选中了也不显示光标栏。</p>
<p>android:digits</p>	<p>设置允许输入哪些字符。如“1234567890.+*%\\n()”</p>
<p>android:drawableTop</p>	<p>在text的正上方输出一个drawable。在TextView中的效果比较搞笑： ，居然在文本框里，而且删不了。</p>
<p>android:drawableBottom</p>	<p>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。</p>
<p>android:drawableLeft</p>	<p>在text的左边输出一个drawable（如图片）。</p>
<p>android:drawablePadding</p>	<p>设置text与drawable(图片)的间隔，与 <code>drawableLeft</code>、<code>drawableRight</code>、<code>drawableTop</code>、<code>drawableBottom</code>一起使用，可设置为负数，单独使用没有效果。</p>
<p>android:drawableRight</p>	<p>在text的右边输出一个drawable，如图片。</p>

<code>android:editable</code>	设置是否可编辑。仍然可以获取光标，但是无法输入。
<code>android:editorExtras</code>	指定特定输入法的扩展，如“com.mydomain.im.SOME_FIELD”。源码跟踪至EditorInfo.extras，暂无相关实现代码。
<code>android:ellipsize</code>	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)
<code>android:freezesText</code>	设置保存文本的内容以及光标的位置。参见： 这里 。
<code>android:gravity</code>	设置文本位置，如设置成“center”，文本将居中显示。
<code>android:hint</code>	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。
<code>android:imeOptions</code>	设置软键盘的Enter键。有如下值可设置： normal, actionUnspecified, actionNone, actionGo , actionSearch, actionSend, actionNext, actionDone , flagNoExtractUi, flagNoAccessoryAction, flagNoEnterAction。可用' '设置多个。这里仅设置显示图标之用，参见文章末尾例子。
<code>android:imeActionId</code>	设置IME动作ID，在onEditorAction中捕获判断进行逻辑操作。
<code>android:imeActionLabel</code>	设置IME动作标签。但是不能保证一定会使用，猜想在输入法扩展的时候应该有用。
<code>android:includeFontPadding</code>	设置文本是否包含顶部和底部额外空白，默认为true。

<p>android:inputMethod</p>	<p>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。关于自定义输入法参见这里。</p>
<p>android:inputType</p>	<p>设置文本的类型，用于帮助输入法显示合适的键盘类型。有如下值设置：none、text、textCapCharacters字母大小、textCapWords单词首字母大小、textCapSentences仅第一个字母大小、textAutoCorrect、textAutoComplete自动完成、textMultiLine多行输入、textImeMultiLine输入法多行（如果支持）、textNoSuggestions不提示、textEmailAddress电子邮件地址、textEmailSubject邮件主题、textShortMessage短信息（会多一个表情按钮出来，点开如下图：</p>  <p>）、textLongMessage长讯息？、textPersonName人名、textPostalAddress地址、textPassword密码、textVisiblePassword可见密码、textWebEditText作为网页表单的文本、textFilter文本筛选过滤、textPhonetic拼音输入、numberSigned有符号数字格式、numberDecimal可带小数点的浮点格式、phone电话号码、datetime时间日期、date日期、time时间。部分参考这里。</p>
<p>android:marqueeRepeatLimit</p>	<p>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。</p>

<code>android:ems</code>	设置TextView的宽度为N个字符的宽度。参见TextView中此属性的截图。
<code>android:maxEms</code>	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。
<code>android:minEms</code>	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。
<code>android:maxLength</code>	限制输入字符数。如设置为5，那么仅可以输入5个汉字/数字/英文字母。
<code>android:lines</code>	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。
<code>android:maxLines</code>	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。
<code>android:minLines</code>	设置文本的最小行数，与lines类似。
<code>android:linksClickable</code>	设置链接是否点击连接，即使设置了autoLink。
<code>android:lineSpacingExtra</code>	设置行间距。
<code>android:lineSpacingMultiplier</code>	设置行间距的倍数。如” 1.2”
<code>android:numeric</code>	如果被设置，该TextView有一个数字输入法。有如下值设置：integer正整数、signed带符号整数、decimal带小数点浮点数。
<code>android:password</code>	以小点”.”显示文本
<code>android:phoneNumber</code>	设置为电话号码的输入方式。
<code>android:privateImeOptions</code>	提供额外的输入法选项(字符串格式)。依据输入法而决定是否提供，如 这里 所见。自定义输入法继承InputMethodService。 这篇文章 也许有帮助。
<code>android:scrollHorizontally</code>	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。参见TextView中此属性的截图。
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。参见TextView的截图。

<code>android:textSize</code>	设置文字大小，推荐度量单位” sp”，如” 15sp”
<code>android:textStyle</code>	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
<code>android:typeface</code>	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]
<code>android:height</code>	设置文本区域的高度，支持度量单位： px(像素)/dp/sp/in/mm(毫米)
<code>android:maxHeight</code>	设置文本区域的最大高度
<code>android:minHeight</code>	设置文本区域的最小高度
<code>android:width</code>	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 。
<code>android:maxWidth</code>	设置文本区域的最大宽度
<code>android:minWidth</code>	设置文本区域的最小宽度

补充说明：

由于是继承自TextView，所以属性是一样的，但是这里重点补充了输入法相关的属性说明和研究，部分注释也做了相应的调整。

三、例子

android:imeOptions例子

```
<EditText android:id="@+id/txtTest" android:imeOptions="actionGo"
    android:layout_width="100dp" android:layout_height="wrap_content">
</EditText>
((EditText)findViewById(R.id.txtTest)).setOnEditorActionListener
(new TextView.OnEditorActionListener() {
    @Override
    public boolean onEditorAction(TextView v, int actionId,
        KeyEvent event) {
        if (actionId == EditorInfo.IME_ACTION_GO) {
            Toast.makeText(TestActivity.this, "你点了
Go!", Toast.LENGTH_SHORT).show();
        }
        return false;
    }
});
```

[Android2.2 API 中文文档系列（3）](#) ——

AccessibilityService

一、结构

```
java.lang.Object
    ↳ android.content.Context
        ↳ android.content.ContextWrapper
            ↳ android.app.Service
                ↳ android.accessibilityservice
                    .AccessibilityService
```

```
public abstract class AccessibilityService extends Service
```

二、类概述

当AccessibilityEvent事件被启动后AccessibilityService 会接收回调函数运行于后台，这些事件指的是在用户接口间的状态转换，比如，焦点变化，按钮被点击等。一些辅助服务继承于此类并且实现它的抽象方法，像这样的一个服务和其他服务一样在AndroidManifest.xml中被声明但它必须被指定操纵

android.accessibilityservice.AccessibilityService的意图，下面的是一段例子：

```
<service android:name=".MyAccessibilityService">
<intent-filter>
<action android:name="android.accessibilityservice.AccessibilityService" />
</intent-filter>
</service>
```

辅助服务的声明周期只能被系统管理，启动或者停止这个服务必须由明确的用户通过启用或停用设备的设定，在系统通过呼叫onServiceConnected()方法与服务绑定后，这个方法才能被想要执行装载的客户端所重载使用，一个辅助服务通过呼叫setServiceInfo(AccessibilityServiceInfo)方法来设定AccessibilityServiceInfo而配置。你可以在任何时候改变这个服务的配置但最好是在重载方法onServiceConnected(). 中来使用。

一个辅助服务可以在特定的包中注册事件以提供特殊的反馈类型并且当最后一个关联的事件被解除的时候发出明确的超时提醒。

三、通告策略

对于每个回馈类型只有一个辅助服务被通知，服务登记处按顺序被通

知，因此，如果有两个服务为同一个包中的同一反馈类型注册那么第一个会被通知，然而有可能的是，可以为一个给定的反馈类型去把一个服务注册为默认的，这样的话如果没有其他的服务来取代这个事件这个服务就会被呼出使用，换句话说，默认的服务不会与 其他的服务竞争并且不管注册的顺序而被通知。

四、公共方法：

4.1 `abstract void onAccessibilityEvent (AccessibilityEvent event)`

Callback for AccessibilityEvents.

参数 `event` 一个事件

4.2 `public final IBinder onBind (Intent intent)`

实现返回一个内部的辅助接口的实现，子类不能被重写。

参数 `intent` 与服务相绑定的意图，注意其他任何包含在Intent的外部意图将不能在此使用。

返回值 返回一个客户端可以在服务上访问的IBinder。

4.3 `public abstract void onInterrupt ()`

打断辅助反馈内容时呼叫。

五、保护方法：

5.1 `protected void onServiceConnected ()`

这个方法是AccessibilityService声明周期的一部分，在系统成功与服务绑定后才被呼叫，如果用来设定AccessibilityServiceInfo这个方法更为方便。

[Android2.2 API 中文文档系列（4） —— Manifest](#)

一、结构

```
java.lang.Object
↳ android.Manifest
```

```
public final class Manifest extends Object
```

内部类

```
Manifest.permission
```

Manifest.permission_group

二、Manifest.permission的常量

ACCESS_CHECKIN_PROPERTIES	允许在登入数据库的时候读写其中的属性表，并上传改变的值
ACCESS_COARSE_LOCATION	允许应用访问范围(如WIFI)性的定位
ACCESS_FINE_LOCATION	允许应用访问精确(如GPS)性的定位
ACCESS_LOCATION_EXTRA_COMMANDS	允许应访问额外的提供定位的指令
ACCESS_MOCK_LOCATION	允许应用创建用于测试的模拟定位提供者
ACCESS_NETWORK_STATE	允许应用访问网络上的信息
ACCESS_SURFACE_FLINGER	允许应用使用低版本视图的特征
ACCESS_WIFI_STATE	允许应用访问关于Wi-Fi网络的信息
ACCOUNT_MANAGER	允许应用进入帐户认证
AUTHENTICATE_ACCOUNTS	允许应用为ACCOUNT_MANAGER扮演一个帐户认证系统
BATTERY_STATS	允许应用去统计电源信息
BIND_APPWIDGET	允许应用告诉AppWidget哪个应用能够访问该AppWidget的数据
BIND_DEVICE_ADMIN	必须通过关机接收者的请求来确保只有系统能够与之交互
BIND_INPUT_METHOD	必须通过InputMethodService的请求来确保只有系统能够与之绑定
BIND_WALLPAPER	必须通过WallpaperService的请求来确保只有系统能够与之绑定
BLUETOOTH	允许应用去连接蓝牙设备
BLUETOOTH_ADMIN	允许应用找到与之连接的蓝牙设备
BRICK	被请求废止设备(非常危险)
BROADCAST_PACKAGE_REMOVED	允许应用发出一个程序包被移除的广播消息
BROADCAST_SMS	允许应用发出一个收到短信的消息
BROADCAST_STICKY	允许应用发出一个与intent相连的消息

BROADCAST_WAP_PUSH	允许应用发出一个收到WAP PUSH的广播消息
CALL_PHONE	允许应用启动一个用户确认电话被拨打而不通过拨打电话的用户界面的的拨打程序
CALL_PRIVILEGED	允许应用启动一个用户确认电话被拨打而不通过拨打电话的用户界面的的任意号码的拨打，包括紧急号码.
CAMERA	能够启动照相机设备的请求
CHANGE_COMPONENT_ENABLED_STATE	允许应用去改变一个应用是否是激活状态
CHANGE_CONFIGURATION	允许应用修改当前的配置，如本地设置
CHANGE_NETWORK_STATE	允许应用改变网络的连接状态
CHANGE_WIFI_MULTICAST_STATE	允许应用进入Wi-Fi的组播方式
CHANGE_WIFI_STATE	允许应用改变Wi-Fi的连接状态
CLEAR_APP_CACHE	允许应用清除所有安装在设备上的应用的缓存
CLEAR_APP_USER_DATA	允许应用清除使用者的信息资料
CONTROL_LOCATION_UPDATES	允许从广播设备来更新或不更新本地的消息
DELETE_CACHE_FILES	允许应用删除掉缓存文件
DELETE_PACKAGES	允许应用删除掉程序包
DEVICE_POWER	允许低权限的访问电源管理项
DIAGNOSTIC	允许应用诊断程序资源
DISABLE_KEYGUARD	允许应用禁用键盘锁
DUMP	允许应用从系统服务中恢复转储的信息
EXPAND_STATUS_BAR	允许应用扩大或缩小状态栏
FACTORY_TEST	如制造商测试的应用一样用终极权限用户运行
FLASHLIGHT	允许访问手电筒
FORCE_BACK	允许应用强制的返回操作而不论是不是最终的activity
GET_ACCOUNTS	允许应用访问账目服务中的统计清单
GET_PACKAGE_SIZE	允许应用查找出任何程序包使用的空间

GET_TASKS	允许应用找到关于当前或最近运行的任务和在某些activities里运行
GLOBAL_SEARCH	这个权限可以被内容提供者用来允许使用全程搜索他们的数据
HARDWARE_TEST	允许访问硬件及周边设备.
INJECT_EVENTS	允许应用注入用户事件（键盘、触摸）到事件中然后提供给任意的窗口
INSTALL_LOCATION_PROVIDER	允许应用安装一个位置提供商到位置管理器中
INSTALL_PACKAGES	允许应用安装程序包.
INTERNAL_SYSTEM_WINDOW	允许应用打开被部分系统用户接口使用的窗口
INTERNET	允许应用打开网络套接口
KILL_BACKGROUND_PROCESSES	允许应用去呼叫killBackgroundProcesses (String). 方法
MANAGE_ACCOUNTS	允许应用去管理帐户管理者中的重要清单
MANAGE_APP_TOKENS	允许应用去管理(创建、销毁、顺序)在窗口管理者中的应用
MASTER_CLEAR	
MODIFY_AUDIO_SETTINGS	允许应用修改全局音频设定
MODIFY_PHONE_STATE	允许改变拨打电话的状态-电源等
MOUNT_FORMAT_FILESYSTEMS	允许格式化可移除的存储仓库的文件系统
MOUNT_UNMOUNT_FILESYSTEMS	允许装备或解除可移除的存储仓库的文件系统
PERSISTENT_ACTIVITY	允许应用使它的activities更持久稳固
PROCESS_OUTGOING_CALLS	允许应用监督、限定或终止呼出的电话
READ_CALENDAR	允许应用读取用户的日历数据
READ_CONTACTS	允许应用读取用户的联系人数据
READ_FRAME_BUFFER	允许应用抓取屏幕和更多可获得的缓冲数据
READ_HISTORY_BOOKMARKS	允许应用去读取(非写)用户浏览历史和书签
READ_INPUT_STATE	允许应用去的当前键盘和控制的状态
READ_LOGS	允许应用读取低级别的系统日志文件

READ_OWNER_DATA	允许应用读取所有者的数据
READ_PHONE_STATE	允许读取电话的状态
READ_SMS	允许应用读取短信息.
READ_SYNC_SETTINGS	允许应用读取同步的设置
READ_SYNC_STATS	允许应用读取同步的统计数据
REBOOT	重新启动设备的请求
RECEIVE_BOOT_COMPLETED	允许应用接收在系统完成启动后发出的ACTION_BOOT_COMPLETED广播信息
RECEIVE_MMS	允许应用去监听多媒体信息并记录和对起进行处理
RECEIVE_SMS	允许应用去监听短消息并记录和对起进行处理
RECEIVE_WAP_PUSH	允许应用监听WAP push信息
RECORD_AUDIO	允许应用记录音频信息
REORDER_TASKS	允许应用改变任务的关系位置
RESTART_PACKAGES	已废弃使用
SEND_SMS	允许应用发送短消息.
SET_ACTIVITY_WATCHER	允许应用查看和控制activities是怎样在系统中运行的
SET_ALWAYS_FINISH	允许应用去控制当activities被覆盖后是否是立即接触结束
SET_ANIMATION_SCALE	改变动画的比例因子
SET_DEBUG_APP	设置一个应用为调试模式
SET_ORIENTATION	允许低级别的设置屏幕的方向
SET_PREFERRED_APPLICATIONS	已废弃
SET_PROCESS_LIMIT	允许应用设置可以运行的最大数的应用进程
SET_TIME	允许应用设置系统时间
SET_TIME_ZONE	允许应用设置系统时区时间
SET_WALLPAPER	允许应用设置壁纸
SET_WALLPAPER_HINTS	允许应用设置锁定的壁纸

SIGNAL_PERSISTENT_PROCESSES	允许应用发出一个给所有稳定进程信号的请求
STATUS_BAR	允许应用打开、关闭或使状态栏或图标失去作用
SUBSCRIBED_FEEDS_READ	允许应用访问内容提供者的签署认证
SUBSCRIBED_FEEDS_WRITE	
SYSTEM_ALERT_WINDOW	允许应用使用TYPE_SYSTEM_ALERT来打开窗口，并将窗口显示于其他应用的顶端
UPDATE_DEVICE_STATS	允许应用更新设备资料信息
USE_CREDENTIALS	允许应用从管理器得到授权请求
VIBRATE	允许应用访问震动器
WAKE_LOCK	允许使用电源锁定管理以使进程休眠或屏幕变暗
WRITE_APN_SETTINGS	允许应用去写入接入点设置
WRITE_CALENDAR	允许应用写（非读）用户的日历数据
WRITE_CONTACTS	允许应用写（非读）用户的联系人数据
WRITE_EXTERNAL_STORAGE	允许应用写（非读）用户的外部存储器
WRITE_GSERVICES	允许应用修改Google服务地图
WRITE_HISTORY_BOOKMARKS	允许应用写（非读）用户的浏览器历史和书签
WRITE_OWNER_DATA	允许应用写（非读）用户的数据
WRITE_SECURE_SETTINGS	允许应用写或读当前系统设置
WRITE_SETTINGS	允许应用写或读系统设置
WRITE_SMS	允许应用写短消息信息
WRITE_SYNC_SETTINGS	允许应用写同步设置

三、Manifest.permission_group的常量

ACCOUNTS	直接通过统计管理器访问管理的统计
COST_MONEY	可以用来让用户花钱但不需要通过与他们直接牵涉的权限

DEVELOPMENT_TOOLS	与开发联盟特征相连的权限组
HARDWARE_CONTROLS	被用来提供直接访问硬件设备的权限
LOCATION	用来允许访问用户的当前位置的权限
MESSAGES	用来允许应用发送用户收到的被拦截的信息
NETWORK	用来提供访问网络服务的权限
PERSONAL_INFO	用于提供访问用户私人数据如联系人、日历、电子邮件等的权限
PHONE_CALLS	用于跟访问和修改拨号状态如截取去话信息、读取和修改电话状态等的权限
STORAGE	与SD卡访问相关联的权限组
SYSTEM_TOOLS	与系统API有关联的权限组

[Android 2.2 API 中文文档系列 \(5\) —— View](#)

一、结构

```
java.lang.Object
    ↳ android.view.View
```

已知直接子类:

AnalogClock, ImageView, KeyboardView, ProgressBar, SurfaceView, [TextView](#), ViewGroup, ViewStub

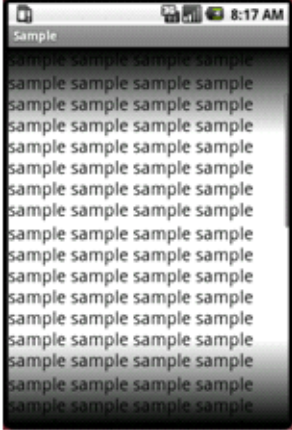
已知间接子类:

AbsListView, AbsSeekBar, AbsSpinner, AbsoluteLayout, AdapterView<T extends Adapter>, AppWidgetHostView, AutoCompleteTextView, Button, CheckBox, CheckedTextView, Chronometer, CompoundButton, DatePicker, DialerFilter, DigitalClock, [EditText](#), ExpandableListView, ExtractEditText, FrameLayout, GLSurfaceView, Gallery, GestureOverlayView, GridView, HorizontalScrollView, ImageButton, ImageSwitcher, LinearLayout, ListView, MediaController, MultiAutoCompleteTextView, QuickContactBadge, RadioButton, RadioGroup, RatingBar, RelativeLayout, ScrollView, SeekBar, SlidingDrawer, Spinner, TabHost, TabWidget, TableLayout, TableRow, TextSwitcher, TimePicker, ToggleButton, TwoLineListItem, VideoView, ViewAnimator, ViewFlipper, ViewSwitcher, WebView, ZoomButton, ZoomControls

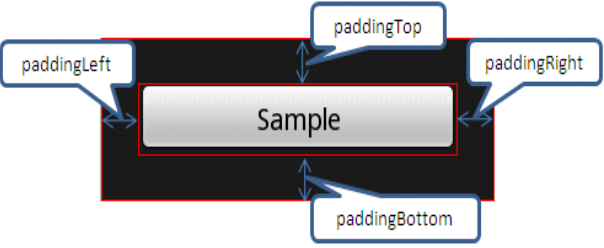
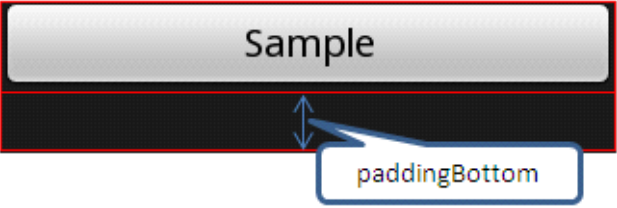


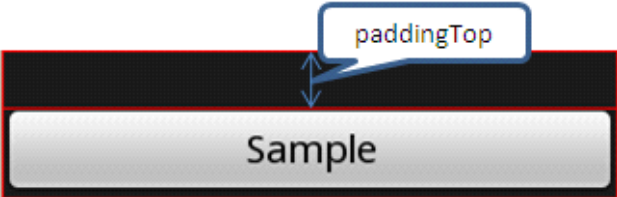
二、xml属性

属性名称	描述
------	----

<p>android:background</p>	<p>设置背景色/背景图片。可以通过以下两种方法设置背景为透明：“@android:color/transparent”和“@null”。注意TextView默认是透明的，不用写此属性，但是Button/ImageButton/ImageView想透明的话就得写这个属性了。</p>
<p>android:clickable</p>	<p>是否响应点击事件。</p>
<p>android:contentDescription</p>	<p>设置View的备注说明，作为一种辅助功能提供，为一些没有文字描述的View提供说明，如ImageButton。这里在界面上不会有效果，自己在程序中控制，可临时放一点字符串数据。</p>
<p>android:drawingCacheQuality</p>	<p>设置绘图时半透明质量。有以下值可设置：auto（默认，由框架决定）/high（高质量，使用较高的颜色深度，消耗更多的内存）/low（低质量，使用较低的颜色深度，但是用更少的内存）。</p> 
<p>android:duplicateParentState</p>	<p>如果设置此属性，将直接从父容器中获取绘图状态（光标，按下等）。见下面代码部分，注意根据目前测试情况仅仅是获取绘图状态，而没有获取事件，也就是你点一下LinearLayout时Button有被点击的效果，但是不执行点击事件。</p>
<p>android:fadingEdge</p>	<p>设置拉滚动条时，边框渐变的放向。none（边框颜色不变），horizontal（水平方向颜色变淡），vertical（垂直方向颜色变淡）。参照fadingEdgeLength的效果图</p>

<p>android:fadingEdgeLength</p>	 <p>设置边框渐变的长度。</p>
<p>android:fitsSystemWindows</p>	<p>设置布局调整时是否考虑系统窗口（如状态栏）</p>
<p>android:focusable</p>	<p>设置是否获得焦点。若有requestFocus()被调用时，后者优先处理。注意在表单中想设置某一个如EditText获取焦点，光设置这个是不行的，需要将这个EditText前面的focusable都设置为false才行。在Touch模式下获取焦点需要设置focusableInTouchMode为true。</p>
<p>android:focusableInTouchMode</p>	<p>设置在Touch模式下View是否能取得焦点。</p>
<p>android:hapticFeedbackEnabled</p>	<p>设置长按时是否接受其他触摸反馈事件。这里模拟器没有试出效果，难道是多点触摸？找不到资料可以找找performHapticFeedback或HapticFeedback这个关键字的资料看看。</p>
<p>android:id</p>	<p>给当前View设置一个在当前layout.xml中的唯一编号，可以通过调用View.findViewById()或Activity.findViewById()根据这个编号查找到对应的View。不同的layout.xml之间定义相同的id不会冲突。格式如”@+id/btnName”</p>
<p>android:isScrollContainer</p>	<p>设置当前View为滚动容器。这里没有测试出效果来，ListView/ GridView/ ScrollView根本就不用设置这个属性，而EdidText设置android:scrollbars也能出滚动条。</p>

<p>android:keepScreenOn</p>	<p>View在可见的情况下是否保持唤醒状态。常在LinearLayout使用该属性，但是模拟器这里没有效果。</p> 
<p>android:longClickable</p>	<p>设置是否响应长按事件.</p>
<p>android:minHeight</p>	<p>设置视图最小高度</p>
<p>android:minWidth</p>	<p>设置视图最小宽度度</p>
<p>android:nextFocusDown</p>	<p>设置下方指定视图获得下一个焦点。焦点移动是基于一个在给定方向查找最近邻居的算法。如果指定视图不存在，移动焦点时将报运行时错误。可以设置imeOptions= actionDone，这样输入完即跳到下一个焦点。</p>
<p>android:nextFocusLeft</p>	<p>设置左边指定视图获得下一个焦点。</p>
<p>android:nextFocusRight</p>	<p>设置右边指定视图获得下一个焦点。</p>
<p>android:nextFocusUp</p>	<p>设置上方指定视图获得下一个焦点。</p>
<p>android:onClick</p>	<p>点击时从上下文中调用指定的方法。这里指定一个方法名称，一般在Activity定义符合如下参数和返回值的函数并将方法名字符串指定为该值即可：</p> <pre>public void onClickButton(View view)</pre> <p>android:onClick=" onClickButton"</p>

<p><code>android:padding</code></p>	<p>设置上下左右的边距，以像素为单位填充空白。</p> 
<p><code>android:paddingBottom</code></p>	<p>设置底部的边距，以像素为单位填充空白。</p> 
<p><code>android:paddingLeft</code></p>	<p>设置左边的边距，以像素为单位填充空白。</p> 
<p><code>android:paddingRight</code></p>	<p>设置右边的边距，以像素为单位填充空白。</p> 
<p><code>android:paddingTop</code></p>	<p>设置上方的边距，以像素为单位填充空白。</p> 

android:saveEnabled	设置是否在窗口冻结时（如旋转屏幕）保存View的数据，默认为true，但是前提是你需要设置id才能自动保存，参见 这里 。
android:scrollX	以像素为单位设置水平方向滚动的偏移值，在GridView中可看的这个效果。
android:scrollY	以像素为单位设置垂直方向滚动的偏移值
android:scrollbarAlwaysDrawHorizontalTrack	设置是否始终显示垂直滚动条。这里用ScrollView、ListView测试均没有效果。
android:scrollbarAlwaysDrawVerticalTrack	设置是否始终显示垂直滚动条。这里用ScrollView、ListView测试均没有效果。
android:scrollbarDefaultDelayBeforeFade	设置N毫秒后开始淡化，以毫秒为单位。
android:scrollbarFadeDuration	设置滚动条淡出效果（从有到慢慢的变淡直至消失）时间，以毫秒为单位。Android2.2中滚动条滚动完之后会消失，再滚动又会出来，在1.5、1.6版本里面会一直显示着。
android:scrollbarSize	设置滚动条的宽度。

<p><code>android:scrollbarStyle</code></p>	<p>设置滚动条的风格和位置。设置值： insideOverlay、insideInset、 outsideOverlay、outsideInset。这里没有试出 太多效果，以下依次是outsideOverlay与 outsideInset效果截图比较：</p> 
<p><code>android:scrollbarThumbHorizontal</code></p>	<p>设置水平滚动条的drawable。</p> 

<p><code>android:scrollbarThumbVertical</code></p>	<p>设置垂直滚动条的drawable.</p> 
<p><code>android:scrollbarTrackHorizontal</code></p>	<p>设置水平滚动条背景（轨迹）的色drawable</p> 

<p><code>android:scrollbarTrackVertical</code></p>	<p>设置垂直滚动条背景（轨迹）的drawable注意直接设置颜色值如” android:color/white” 将得出很难看的效果，甚至都不理解这个属性了，这里可以参见ApiDemos里res/drawable/scrollbar_vertical_thumb.xml和scrollbar_vertical_track.xml，设置代码为：<code>android:scrollbarTrackVertical="@drawable/scrollbar_vertical_track”</code></p> 
<p><code>android:scrollbars</code></p>	<p>设置滚动条显示。none（隐藏），horizontal（水平），vertical（垂直）。见下列代码演示使用该属性让EditText内有滚动条。但是其他容器如LinearLayout设置了但是没有效果。</p> 
<p><code>android:soundEffectsEnabled</code></p>	<p>设置点击或触摸时是否有声音效果</p>

android:tag	<p>设置一个文本标签。可以通过View.getTag()或for with</p> <p>View.findViewById()检索含有该标签字符串的View。但</p> <p>一般最好通过ID来查询View，因为它的速度更快，并且允许编译时类型检查。</p>
android:visibility	<p>设置是否显示View。设置值：visible（默认值，显示），invisible（不显示，但是仍然占用空间），gone（不显示，不占用空间）</p>

三、代码

3.1 android:duplicateParentState

```
<LinearLayout android:clickable="true" android:background="#ff0fff" android:layout_width="100dp" android:layout_height="100dp">
    <Button android:duplicateParentState="true" android:layout_width="wrap_content" android:layout_height="wrap_content"/>
</LinearLayout>
```

3.2 android:scrollbars

```
<EditText android:layout_width="fill_parent"
android:layout_height="wrap_content" android:minHeight="50dp"
android:background="@android:drawable/editbox_background"
android:scrollbars="vertical"
    android:maxLines="4">
</EditText>
```

四、遗留问题

4.1 以下几个属性翻遍了资料试了很多次都没有效果，只能暂时搁置，以后补上，也欢迎大家提供意见和线索，分享大家的经验：

```
android:scrollbarAlwaysDrawHorizontalTrack
android:scrollbarAlwaysDrawVerticalTrack
android:isScrollContainer
```

[Android2.2 API 中文文档系列（6） —— ImageView](#)

一、结构

java.lang.Object

↳ [android.view.View](#)

↳ android.widget.ImageView

已知直接子类:

ImageButton, QuickContactBadge


已知间接子类:

ZoomButton

二、类概述


显示任意图像，例如图标。ImageView类可以加载各种来源的图片（如资源或图片库），需要计算图像的尺寸，以便它可以在其他布局中使用，并提供例如缩放和着色（渲染）各种显示选项。

三、XML属性

属性名称	描述
android:adjustViewBounds	是否保持宽高比。需要与maxWidth、MaxHeight一起使用，否则单独使用没有效果。
android:cropToPadding	是否截取指定区域用空白代替。单独设置无效果，需要与scrollY一起使用，效果如下，实现代码见代码部分： 

<code>android:maxHeight</code>	<p>设置View的最大高度，单独使用无效，需要与setAdjustViewBounds一起使用。如果想设置图片固定大小，又想保持图片宽高比，需要如下设置：</p> <ol style="list-style-type: none">1) 设置setAdjustViewBounds为true;2) 设置maxWidth、MaxHeight;3) 设置设置layout_width和layout_height为wrap_content。
<code>android:maxWidth</code>	设置View的最大宽度。同上。

设置图片的填充方式。

matrix	0	用矩阵来绘图	
fitXY	1	拉伸图片（不按比例）以填充View的宽高 	layout - height :30px
fitStart	2	按比例拉伸图片，拉伸后图片的高度为View的高度，且显示在View的左边 	layout - width :120px
fitCenter	3	按比例拉伸图片，拉伸后图片的高度为View的高度，且显示在View的中间 	
fitEnd	4	按比例拉伸图片，拉伸后图片的高度为View的高度，且显示在View的右边 	
center	5	按原图大小显示图片，但图片宽高大于View的宽高时，截图图片中间部分显示 	layout - height :60px
31 centerCrop	6	按比例放大原图直至等于某边View的宽高显 	layout - width :80px paddin

android:scaleType

android:src	设置View的drawable(如图片, 也可以是颜色, 但是需要指定View的大小)
android:tint	<p>将图片渲染成指定的颜色。见下图:</p>  <p>左边为原图, 右边为设置后的效果, 见后面代码。</p>

四、代码

4.1 android:tint

```
<ImageView android:background="@android:color/white" android:src="@drawable/btn_mode_switch_bg"
    android:layout_width="wrap_content" android:layout_height="wrap_content"></ImageView>
    <ImageView android:layout_marginLeft="5dp" android:background="@android:color/white" android:tint="#ffff00" android:src="@drawable/btn_mode_switch_bg"
    android:layout_width="wrap_content" android:layout_height="wrap_content"></ImageView>
```

4.2 android:cropToPadding

```
<ImageView android:background="@android:color/white" android:scrollY="10px" android:cropToPadding="true" android:src="@drawable/btn_mode_switch_bg"
    android:layout_width="wrap_content" android:layout_height="wrap_content"></ImageView>
    <ImageView android:background="@android:color/white" android:scrollY="10px" android:cropToPadding="true" android:src="@drawable/btn_mode_switch_bg"
    android:layout_width="wrap_content" android:layout_height="wrap_content"></ImageView>
    <ImageView android:paddingTop="10px" android:background="@android:color/white" android:scrollY="10px" android:cropToPadding="true" android:src="@drawable/btn_mode_switch_bg"
    android:layout_width="wrap_content" android:layout_height="wrap_content"></ImageView>
    <ImageView android:paddingTop="10px" android:background="@android:color/white" android:scrollY="10px" android:cropToPadding="true" android:src="@drawable/btn_mode_switch_bg"
    android:layout_width="wrap_content" android:layout_height="wrap_content"></ImageView>
```



```
:color/white" android:scrollY="10px" android:cropToPadding="false" android:src="@drawable/btn_mode_switch_bg"
    android:layout_width="wrap_content" android:layout_height="wrap_content"></ImageView>
```

[Android 2.2 API 中文文档系列 \(7\) —— ImageButton](#)

一、结构

```
java.lang.Object
├── android.view.View
│   └── android.widget.ImageView
│       └── android.widget.ImageButton
```

已知直接子类:



ZoomButton

二、类摘要

显示一个可以被用户点击图片按钮，默认情况下，ImageButton看起来像一个普通的按钮，在不同状态（如按下）下改变背景颜色。按钮的图片可用通过<ImageButton> XML元素的android:src属性或setImageResource(int)方法指定。

要删除按钮的背景，可以定义自己的背景图片或设置背景为透明。

(注：请看

原图  和图片按钮 ，默认图片周围有按钮的背景，选中之后为黄色)

为了表示不同的按钮状态（焦点，选择等），你可以为各种状态定义不同的图片。例如，定义蓝色图片为默认图片，黄色图片为获取时焦点时显示的图片，黄色图片为按钮被按下时显示的图片。一个简单的方法可以做到这一点——通过XML的“selector.”配置，如下：

```

<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:state_pressed="true"
        android:drawable="@drawable/button_pressed" /> <!-- pressed -->
  <item android:state_focused="true"
        android:drawable="@drawable/button_focused" /> <!-- focused -->
  <item android:drawable="@drawable/button_normal" /> <!-- default -->
</selector>

```

保存上面的XML到res/drawable/文件夹下（注：注意文件名大小写！），将该文件名作为一个参数设置到ImageButton的 android:src属性（注：如xml文件名为myselector.xml，那么这里设置为"@drawable/myselector"，设置android:background也是可以的，但效果不太一样）。Android根据按钮的状态改变会自动的去XML中查找相应的图片以显示。

<item>元素的顺序很重要，因为是根据这个顺序判断是否适用于当前按钮状态，这也是为什么正常（默认）状态指定的图片放在最后，是因为它只会在pressed和focused都判断失败之后才会被采用。（注：例如按钮被按下时是同时获得焦点的，但是获得焦点并不一定按了按钮，所以这里会按顺序查找，找到合适的就不往下找了。这里按钮被点击了，那么第一个将被选中，且不再在后面查找其他状态。）

参见[Form Stuff tutorial](#)。

三、继承自父类的方法

```
public void setAlpha (int alpha)
```

设置ImageButton图片的透明度（注意不是背景图片的）。效果如图：

参数

alpha 透明值0~255，0为完全透明，255为完全不透明

