



Android总结

作者: iaiai <http://iaiai.iteye.com>

丸子博客android文章精选
生成日期:2011-8-23
QQ:176291935

目 录

1. android基础知识

1.1 Android控件之ListView探究一	7
1.2 Android控件之ListView探究二，带CheckBox的ListView	14
1.3 android布局 LinearLayout和RelativeLayout	22
1.4 Android控件之Menu选择菜单	25
1.5 Android控件之AutoCompleteTextView、MultiAutoCompleteTextView探究	28
1.6 Android控件之DatePicker、TimePicker探究	31
1.7 Android控件之TextView探究	35
1.8 Android控件之EditText探究	36
1.9 Android Activity之间跳转相互传值	37
1.10 Android之Intent探究	45
1.11 Android 电话拨号	49
1.12 Android 短信发送器	54
1.13 Android 中的显示单位	58
1.14 Android 使用文件进行数据存储	59
1.15 Android 把文件存放在SDCard	64
1.16 Android SQLite数据库增删改查操作	68
1.17 Android 获取联系人	75
1.18 Android 进行单元测试	84
1.19 Android 获得本机ip地址和MAC地址	86

1.20 Android 事件监听	89
1.21 Android 控件使用参数集锦	99
1.22 Android HttpClient网络通信	105
1.23 Android 读取SIM卡参数	110
1.24 Android之permission权限列表	120
1.25 Android开发中如何实现开机自启动	124
1.26 Android Activity之间切换的各种效果	125
1.27 Android 设置横屏/竖屏	134
1.28 Android之快捷方式——通过应用程序创建快捷方式	138
1.29 Android之快捷方式二——向Launcher添加快捷方式	140
1.30 Android中实现全屏显示的方法	143
1.31 Android SQLite数据库实例	145
1.32 Android listView 一种常用布局	148
1.33 Android md5加密	150
1.34 Android 画图之Matrix(一)	151
1.35 Android 画图之Matrix(二)	154
1.36 Android 多式样ProgressBar	157
1.37 Android 边框圆角	159
1.38 Android 背景渐变色	161
1.39 Android Widget开发	163
1.40 Android实现iphone风格的picker	169
1.41 Android 控件之SeekBar	173

1.42 Android 控件之切换卡(TabWidget)	177
1.43 Android 标题栏(titlebar)显示进度条	182
1.44 Android 控件之Notification	183
1.45 Android 控件之Service并开机自动启动	189
1.46 Android 拖动换图特效	196
1.47 Android 使用Gallery实现Tab	200
1.48 Android ListView的分段显示、分页显示(一)	206
1.49 Android ListView的分段显示、分页显示(二)	211
1.50 Android sqllite中日期时间问题	216
1.51 Android开发环境搭建全程演示	217
1.52 Android开发配置需要注意的地方	235
1.53 Android画图之抗锯齿	237
1.54 Android 组件 SeekBar,RatingBar,Chronometer	240
1.55 Android之SharedPreferences	247
1.56 Android之TableLayout	250
1.57 Android之GridView(九宫图)	252
1.58 android 屏幕分辨率问题	256
1.59 Android中Activity共享变量的另一方法 : Application context	262
1.60 Android布局文件中的include	265
1.61 Android屏幕分辨率详解 (VGA、HVGA、QVGA、WVGA、WQVGA)	267
1.62 Android 2.2完全退出程序, 使用广播机制	268
1.63 Android 拖拽图片	270

1.64 Android控件之Gallery实现图片列表	272
1.65 Android XML文件使用	278
1.66 Android中尺寸单位杂谈	286
1.67 Android系统中的.apk文件和dex文件	288
1.68 Android 调用系统摄像头	291
1.69 Android 自定义ListView背景	293
1.70 Android frame by frame animation动画显示	295
1.71 Android HandlerThread类 , Bundle的使用	300
1.72 Android Launcher研究	303
1.73 Android 图片加圆角以及时间处理工具类	304
1.74 Android浏览器显示大分辨率图片的问题 分享	310
1.75 Android获取其他包的Context实例然后做坏事	311
1.76 Android有效解决加载大图片时内存溢出的问题	313
1.77 Android善用预定义样式	318
1.78 Android高手应该精通哪些内容?	320
1.79 Activity 间传递对象, 采用Parcelable	321
1.80 Android NDK简介	326
1.81 Android 总结4种线程中操作UI界面的方法	327
1.82 Android获取状态栏和标题栏的高度	330
1.83 Android 获取手机屏幕大小两种方法	331
1.84 android:clearTaskOnLaunch="true"什么意思 ?	332
1.85 Android TabSwitcher自定义控件	333

1.86 Android 左右滑动 控件	340
1.87 Android中利用画图类和线程画出闪烁的心形，送给亲爱的他	361
1.88 Android PreferenceActivity用法	369
1.89 Android 9 PATCH 图片	372
1.90 Android 怎么样使用shape	373
1.91 Android 自定义PopupWindow实现的Menu(TabMenu)	378
1.92 Android 日历方式显示的日期选择组件	391
1.93 Android 多层树完美实现	418
2. android组件(Widget)开发	
2.1 Android 联系人选择Widget	430
2.2 Android QQ登录界面	441
3. android 错误	
3.1 Using default keymap: /system/usr/keychars/qwerty.kcm.bin	446
3.2 Error inflating class xxx.xxx.xxx	447
3.3 Parcelable encountered IOException writing serializable object (name = xxx)	448
4. 工具(IDE)/包(jar)	
4.1 android NDK下载地址	450
4.2 Android API	451

1.1 Android控件之ListView探究一

发表时间: 2011-05-28 关键字: Android, Java, ITeye, XML, OS

在android开发中ListView是比较常用的组件，它以列表的形式展示具体内容，并且能够根据数据的长度自适应显示。

main.xml布局文件

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <ListView android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:id="@+id/MyListView">
    </ListView>
</LinearLayout>
```

list_items.xml布局文件

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:id="@+id/myListItem"
    android:paddingBottom="4dip" android:paddingLeft="12dip"
    android:paddingRight="12dip">
    <TextView android:layout_height="wrap_content"
        android:layout_width="fill_parent" android:id="@+id/itemTitle"
        android:textSize="20dip" android:layout_weight="1">
    </TextView>
    <TextView android:layout_height="wrap_content"
        android:layout_width="fill_parent" android:id="@+id/itemText">
    </TextView>
</LinearLayout>
```

MyList.java类文件


```
        new String[] { "itemTitle", "itemText" }, // 数据源的属性字段
        new int[] { R.id.itemTitle, R.id.itemText }); // 布局里的控件id
    // 添加并且显示
    list.setAdapter(adapter);

    // 添加点击
    list.setOnItemClickListener(new OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
                long arg3) {
            setTitle("点击第" + arg2 + "个项目");
            Toast.makeText(MyList.this, "点击第" + arg2 + "个项目",
                    Toast.LENGTH_LONG).show();
        }
    });

    // 添加长按点击
    list.setOnCreateContextMenuListener(new OnCreateContextMenuListener() {
        @Override
        public void onCreateContextMenu(ContextMenu menu, View v,
                ContextMenuInfo menuInfo) {
            // menu.setHeaderIcon(R.drawable.icon); //字义图片样式
            menu.setHeaderTitle("长按菜单选项:");
            menu.add(1, 0, 0, "编辑");
            menu.add(0, 1, 0, "删除");
        }
    });
}

@Override
public boolean onContextItemSelected(MenuItem item) {
    int selectedPosition = ((AdapterContextMenuInfo) item.getMenuInfo()).position;
    Map<String, String> map = (HashMap) list
            .getItemAtPosition(selectedPosition); // ListView中的数据
    Toast.makeText(MyList.this,
            "你选择的:" + selectedPosition + "，内容为:" + map.get("itemTitl
    Toast.LENGTH_LONG).show(); // 显示那条数据
}
```

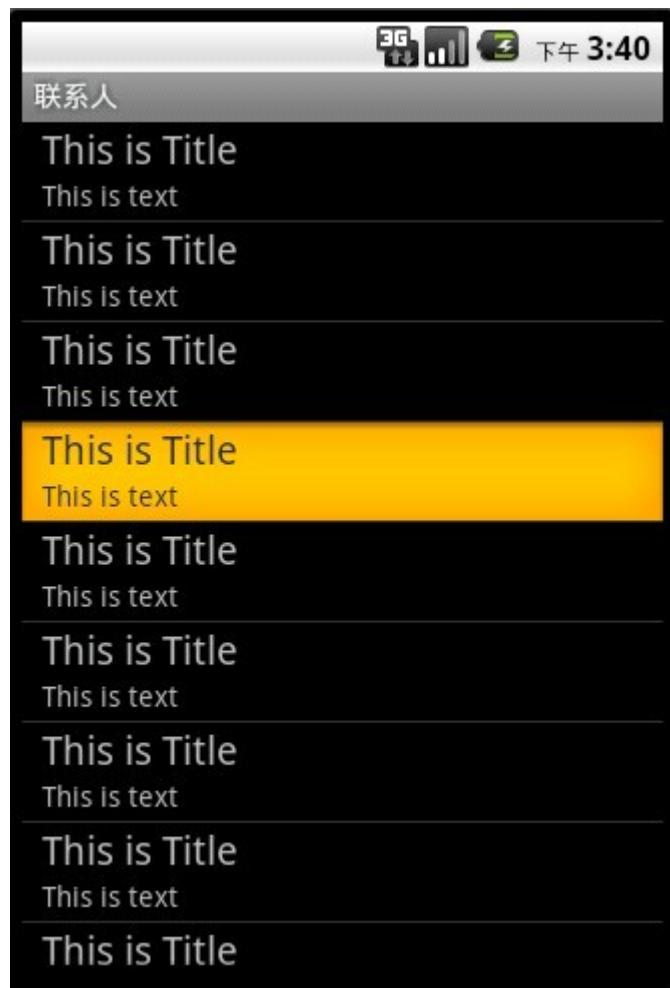
```
String temp = "";
switch (item.getItemId()) {
    case 0:
        temp = "编辑";
        break;
    case 1:
        temp = "删除";
        break;
    default:
        break;
}
// Toast.makeText(this, temp + "處理", Toast.LENGTH_LONG).show();
return super.onContextItemSelected(item);
}

private List<Map<String, String>> getData() {
    // 组织数据源
    List<Map<String, String>> mylist = new ArrayList<Map<String, String>>();
    for (int i = 0; i < 10; i++) {
        Map<String, String> map = new HashMap<String, String>();
        map.put("itemTitle", "文章" + i);
        map.put("itemText", "Android 小项目" + i);
        mylist.add(map);
    }
    return mylist;
}

/**
 * listview中点击按键弹出对话框
 */
public void showInfo() {
    new AlertDialog.Builder(this).setTitle("我的listview")
        .setMessage("介绍...")
        .setPositiveButton("确定", new DialogInterface.OnClickListener(
            @Override
            public void onClick(DialogInterface dialog, int which)
```

```
        }  
    }).show();  
}  
  
}
```

运行结果:





ListView不可放入ScrollView中，否则UI效果不可预测。切记切记

1.2 Android控件之ListView探究二，带CheckBox的ListView

发表时间: 2011-05-30 关键字: Android, Java, ITeye, XML, OS

布局用的是RelativeLayout这种布局方式，要指定位置，不太好，以后如果有其它好的方式再修改

main.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <ListView android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:id="@+id/MyListView">
    </ListView>
</LinearLayout>
```

list_items.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:id="@+id/myListItem"
    android:paddingBottom="5dip" android:paddingLeft="12dip"
    android:paddingRight="12dip">
    <ImageView android:id="@+id/img" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_alignParentLeft="true"></Image
    <TextView android:layout_height="wrap_content"
        android:layout_width="wrap_content" android:id="@+id/itemTitle"
        android:textSize="20dip" android:paddingTop="15px"
        android:layout_marginLeft="70px">
    </TextView>
    <CheckBox android:id="@+id/cb" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_alignParentRight="true"
        android:focusable="false"></CheckBox>
</RelativeLayout>
```

MyList.java类：

```
package iaiai.test;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.ContextMenu;
import android.view.ContextMenu.ContextMenuItemInfo;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnCreateContextMenuListener;
import android.widget.AdapterView;
import android.widget.AdapterView.AdapterContextMenuInfo;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;
import android.widget.Toast;

public class MyList extends Activity {

    private ListView list;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        list = (ListView) findViewById(R.id.MyListView);

        // 配置适配器
        MyCheckBoxAdapter adapter = new MyCheckBoxAdapter(this,getData()); // 布局里的拖
```

```
// 添加并且显示
list.setAdapter(adapter);

// 添加点击
list.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
            long arg3) {
        setTitle("点击第" + arg2 + "个项目");
        Toast.makeText(MyList.this, "点击第" + arg2 + "个项目",
                Toast.LENGTH_LONG).show();
    }
});

// 添加长按点击
list.setOnCreateContextMenuListener(new OnCreateContextMenuListener() {
    @Override
    public void onCreateContextMenu(ContextMenu menu, View v,
            ContextMenuInfo menuInfo) {
        // menu.setHeaderIcon(R.drawable.icon); //字义图片样式
        menu.setHeaderTitle("长按菜单选项:");
        menu.add(1, 0, 0, "编辑");
        menu.add(0, 1, 0, "删除");
    }
});

@Override
public boolean onContextItemSelected(MenuItem item) {
    int selectedPosition = ((AdapterContextMenuInfo) item.getMenuInfo()).position;
    Map<String, String> map = (HashMap) list
            .getItemAtPosition(selectedPosition); // ListView中的数据
    Toast.makeText(MyList.this,
            "你选择的:" + selectedPosition + "，内容为:" + map.get("itemTitl
    Toast.LENGTH_LONG).show(); // 显示那条数据

    String temp = "";
}
```

```
switch (item.getItemId()) {  
    case 0:  
        temp = "编辑";  
        break;  
    case 1:  
        temp = "删除";  
        break;  
    default:  
        break;  
}  
// Toast.makeText(this, temp + "處理", Toast.LENGTH_LONG).show();  
return super.onContextItemSelected(item);  
}  
  
private List<Map<String, String>> getData() {  
    // 组织数据源  
    List<Map<String, String>> mylist = new ArrayList<Map<String, String>>();  
    for (int i = 0; i < 10; i++) {  
        Map<String, String> map = new HashMap<String, String>();  
        map.put(""+ i, i + "");  
        map.put("itemTitle", "文章" + i);  
        mylist.add(map);  
    }  
    return mylist;  
}  
  
/**  
 * listview中点击按键弹出对话框  
 */  
public void showInfo() {  
    new AlertDialog.Builder(this).setTitle("我的listview")  
        .setMessage("介绍...")  
        .setPositiveButton("确定", new DialogInterface.OnClickListener()  
            @Override  
            public void onClick(DialogInterface dialog, int which)  
            {}  
        }).show();  
}
```

```
    }  
}
```

MyCheckBoxAdapter.java类：

```
package iaiai.test;  
  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.View.OnClickListener;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.CheckBox;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
public class MyCheckBoxAdapter extends BaseAdapter {  
  
    private LayoutInflater mInflater;  
  
    private List<Map<String, String>> listData;  
  
    private Map<Integer, Map<String, String>> selectMap = new HashMap<Integer, Map<String,  
  
    private class ViewHolder {  
        public ImageView img;  
        public TextView title;  
        public CheckBox checkBox;  
    }  
}
```

```
public MyCheckBoxAdapter(Context context, List<Map<String, String>> listData) {  
    this.mInflater = LayoutInflater.from(context);  
    this.listData = listData;  
}  
  
@Override  
public int getCount() {  
    return listData.size();  
}  
  
@Override  
public Object getItem(int position) {  
    return listData.get(position);  
}  
  
@Override  
public long getItemId(int position) {  
    return position;  
}  
  
@Override  
public View getView(final int position, View convertView, ViewGroup parent) {  
    ViewHolder holder = null;  
    if (convertView == null) {  
        holder = new ViewHolder();  
        convertView = mInflater.inflate(R.layout.list_items, null);  
        final View view = convertView;  
        holder.img = (ImageView) convertView.findViewById(R.id.img);  
        holder.title = (TextView) convertView.findViewById(R.id.itemTitle);  
        holder.checkBox = (CheckBox) convertView.findViewById(R.id.cb);  
        holder.checkBox.setOnClickListener(new OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                if (selectMap.get(position) != null) {  
                    selectMap.remove(position);  
                } else {  
                    selectMap.put(position, listData.get(position));  
                }  
            }  
        });  
    }  
    return convertView;  
}  
}  
class ViewHolder {  
    ImageView img;  
    TextView title;  
    CheckBox checkBox;  
}
```

```
        }

        Toast.makeText(view.getContext(),
                "你选择了：" + selectMap.size() + "个。",
                Toast.LENGTH_LONG).show();
    }

    convertView.setTag(holder);
}

else {
    holder = (ViewHolder) convertView.getTag();
}

holder.img.setBackgroundResource(R.drawable.icon);
holder.title.setText(listData.get(position).get("itemTitle"));

if (selectMap.get(position) != null) {
    holder.checkBox.setChecked(true);
} else {
    holder.checkBox.setChecked(false);
}

return convertView;
}

}
```

这篇文章主要是学习BaseAdapter类的使用，发的文章都是经过运行可以跑起来的，并且代码全会发到上面。

下面是运行结果：



1.3 android布局 LinearLayout和RelativeLayout

发表时间: 2011-05-30 关键字: Android, XML

常用的布局还有一种TableLayout以后用到了再详细说明，这次先说最用到的两种布局容器。

LinearLayout和RelativeLayout

共有属性：

java代码中通过btn1关联次控件

android:id="@+id btn1"

控件宽度

android:layout_width="80px" // "80dip"或"80dp"

android:layout_width = "wrap_content"

android:layout_width = "match_parent"

控件高度

android:layout_height="80px" // "80dip"或"80dp"

android:layout_height = "wrap_content"

android:layout_height = "match_parent"

控件排布

android:orientation="horizontal"

android:orientation="vertical"

控件间距

android:layout_marginLeft="5dip" //距离左边

android:layout_marginRight="5dip" //距离右边

android:layout_marginTop="5dip" //距离上面

android:layout_marginBottom="5dip" //距离下面

控件显示位置

android:gravity="center" //left,right, top, bottom

android:gravity="center_horizontal"

android:layout_gravity是本元素对父元素的重力方向。

android:layout_gravity属性则设置控件本身相对于父控件的显示位置

android:gravity是本元素所有子元素的重力方向。

```
android:layout_gravity="center_vertical"  
android:layout_gravity="left"  
android:layout_gravity="left|bottom"
```

TextView中文本字体

```
android:text="@String/text1" //在string.xml中定义text1的值  
android:textSize="20sp"  
android:textColor="#ff123456"  
android:textStyle="bold" //普通 ( normal ) , 斜体 ( italic ) , 粗斜体 ( bold_italic )
```

定义控件是否可见

```
android:visibility="visible" //可见  
android:visibility="invisible" //不可见,但是在布局中占用的位置还在  
android:visibility="gone" //不可见,完全从布局中消失
```

定义背景图片

```
android:background="@drawable/img_bg" //img_bg为drawable下的一张图片
```

seekbar控件背景图片及最大值

```
android:progressDrawable="@drawable/seekbar_img"  
android:thumb="@drawable/thumb"  
android:max="60"
```

仅在RelativeLayout中有效：

在父亲布局的相对位置

```
android:layout_alignParentLeft="true" //在布局左边  
android:layout_alignParentRight="true" //在布局右边  
android:layout_alignParentTop="true" //在布局上面  
android:layout_alignParentBottom="true" //在布局的下面
```

在某个控件的相对位置

```
android:layout_toRightOf="@+id/button1" //在控件button1的右边,不仅仅是紧靠着  
android:layout_toLeftOf="@+id/button2" //在控件button2的左边,不仅仅是紧靠着
```

```
android:layout_below="@+id/button1" //在控件button1下面，不仅仅是正下方  
android:layout_above="@+id/button1" //在控件button1上面,不仅仅是正上方
```

定义和某控件对齐

```
android:layout_alignTop="@+id/button1" //和控件button1上对齐  
android:layout_alignBottom="@+id/button1" //和控件button1下对齐  
android:layout_alignLeft="@+id/button1" //和控件button1左对齐  
android:layout_alignRight="@+id/button1" //和控件button2右对齐
```

```
android:layout_centerHorizontal="true" //水平居中  
android:layout_centerVertical="true"  
android:layout_centerInParent="true"
```

仅在LinearLayout中有效

设置控件在一排或一列中所占比例值

```
android:layout_weight="1"
```

1.4 Android控件之Menu选择菜单

发表时间: 2011-05-30 关键字: Android, ITeye, OS

这个Menu选择菜单选择只要在Activity继承类里实现以下方法就可以了

MyList.java类:

```
package iaiai.test;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Toast;

public class MyList extends Activity {

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

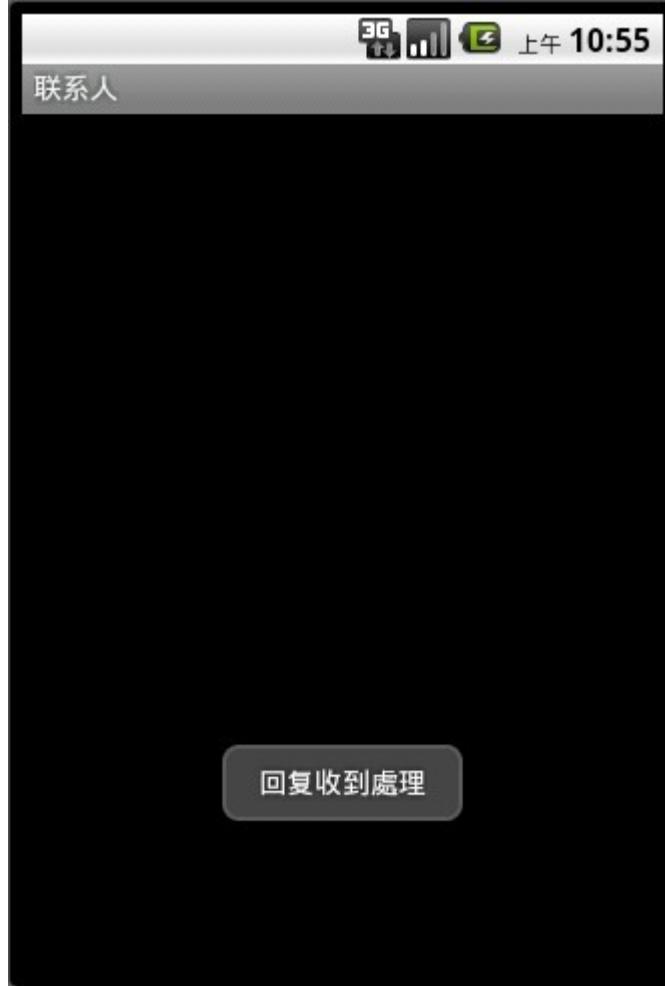
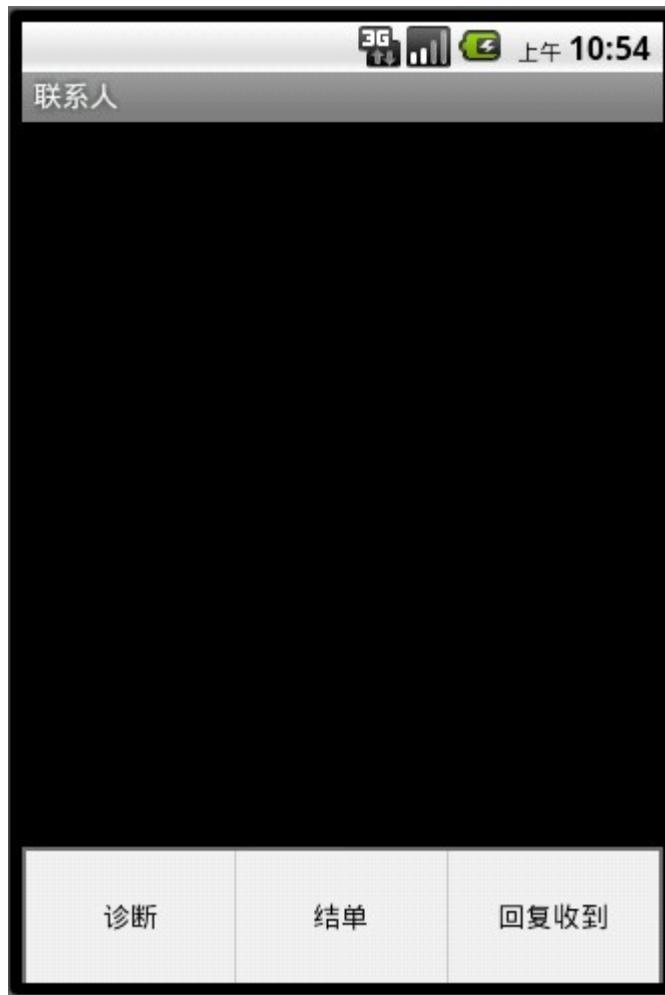
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        menu.add(Menu.NONE, 1, 0, "诊断");
        menu.add(Menu.NONE, 2, 0, "结单");
        menu.add(Menu.NONE, 3, 0, "回复收到");
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public void openOptionsMenu() {
        super.openOptionsMenu();
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
```

```
String temp = "";
switch (item.getItemId()) {
    case 1:
        temp = "诊断";
        break;
    case 2:
        temp = "结单";
        break;
    case 3:
        temp = "回复收到";
        break;
    default:
        break;
}
Toast.makeText(this, temp + "處理", Toast.LENGTH_LONG).show();
return super.onOptionsItemSelected(item);
}

}
```



1.5 Android控件之AutoCompleteTextView、MultiAutoCompleteTextView探究

发表时间: 2011-05-30 关键字: Android, 搜索引擎, Google, ITeye, OS

在Android中提供了两种智能输入框，它们是AutoCompleteTextView、MultiAutoCompleteTextView。它们的功能大致一样。显示效果像Google搜索一样，当你在搜索框里输入一些字符时（至少两个字符），会自动弹出一个下拉框提示类似的结果。下面详细介绍一下。

一、AutoCompleteTextView

1、简介

一个继承自EditText的可编辑的文本视图，能够实现动态匹配输入的内容。如google搜索引擎当输入文本时可以根据内容显示匹配的热门信息。

2、重要方法

clearListSelection()：清除选中的列表项

dismissDropDown()：如果存在关闭下拉菜单

getAdapter()：获取适配器

二、MultiAutoCompleteTextView

1、简介

一个继承自AutoCompleteTextView的可编辑的文本视图，能够对用户键入的文本进行有效地扩充提示，而不需要用户输入整个内容。（用户输入一部分内容，剩下的部分系统就会给予提示）。

用户必须提供一个MultiAutoCompleteTextView.Tokenizer用来区分不同的子串。

2、重要方法

enoughToFilter()：当文本长度超过阈值时过滤

performValidation()：代替验证整个文本，这个子类方法验证每个单独的文字标记

setTokenizer(MultiAutoCompleteTextView.Tokenizer t)：用户正在输入时，tokenizer设置将用于确定文本相关范围内

以下为案例

main.xml布局文件:

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

```
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <AutoCompleteTextView android:id="@+id/autoCompleteTextView"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"/>
    <MultiAutoCompleteTextView android:id="@+id/multiAutoCompleteTextView"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"/>
</LinearLayout>
```

MyList.java类：

```
package iaiai.test;

import android.app.Activity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.widget.MultiAutoCompleteTextView;

public class MyList extends Activity {

    private static final String[] cities = new String[] { "ShiJiaZhuang", "BeiJing", "ShangHai", "GuangZhou", "ChengDu", "XiAn", "NanJing", "WuXi", "HunAn", "HunNan", "TianJin", "BeiJing", "ShangHai", "GuangZhou", "ChengDu", "XiAn", "NanJing", "WuXi", "HunAn", "HunNan", "TianJin" };
    private AutoCompleteTextView autoCompleteTextView = null;
    private MultiAutoCompleteTextView multiAutoCompleteTextView = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        autoCompleteTextView = (AutoCompleteTextView) findViewById(R.id.autoCompleteTextView);
        multiAutoCompleteTextView = (MultiAutoCompleteTextView) findViewById(R.id.multiAutoCompleteTextView);

        // 创建适配器
        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_dropdown_item_1line, cities);
        autoCompleteTextView.setAdapter(adapter);
        multiAutoCompleteTextView.setAdapter(adapter);
    }
}
```

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,  
        android.R.layout.simple_dropdown_item_1line, cities);  
autoCompleteTextView.setAdapter(adapter);  
// 设置输入多少字符后提示，默认值为2  
autoCompleteTextView.setThreshold(2);  
  
multiAutoCompleteTextView.setAdapter(adapter);  
multiAutoCompleteTextView.setThreshold(2);  
// 用户必须提供一个MultiAutoCompleteTextView.Tokenizer用来区分不同的子串。  
multiAutoCompleteTextView  
        .setTokenizer(new MultiAutoCompleteTextView.CommaTokenizer());  
}  
  
}
```

运行结果:



1.6 Android控件之DatePicker、TimePicker探究

发表时间: 2011-05-30 关键字: Android, ITeye, OS, XML

一、 DatePicker继承自FrameLayout类，日期选择控件的主要功能是向用户提供包含年、月、日的日期数据并允许用户对其修改。如果要捕获用户修改日期选择控件中的数据事件，需要为DatePicker添加OnDateChangedListener监听器。

二、 TimePicker也继承自FrameLayout类。时间选择控件向用户显示一天中的时间（可以为24小时，也可以为AM/PM制），并允许用户进行选择。如果要捕获用户修改时间数据的事件，便需要为TimePicker添加OnTimeChangedListener监听器

以下模拟日期与时间选择控件的用法

main.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <DatePicker android:id="@+id/datePicker"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal" />
    <EditText android:id="@+id/dateEt" android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:cursorVisible="false"
        android:editable="false" />
    <TimePicker android:id="@+id/timePicker"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal" />
    <EditText android:id="@+id/timeEt" android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:cursorVisible="false"
        android:editable="false" />
</LinearLayout>
```

MyList.java类:

```
package iaiai.test;

import java.util.Calendar;

import android.app.Activity;
import android.os.Bundle;
import android.widget.DatePicker;
import android.widget.DatePicker.OnDateChangedListener;
import android.widget.EditText;
import android.widget.TimePicker;
import android.widget.TimePicker.OnTimeChangedListener;

public class MyList extends Activity {

    private EditText dateEt;
    private EditText timeEt;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        dateEt = (EditText) findViewById(R.id.dateEt);
        timeEt = (EditText) findViewById(R.id.timeEt);
        DatePicker datePicker = (DatePicker) findViewById(R.id.datePicker);
        TimePicker timePicker = (TimePicker) findViewById(R.id.timePicker);

        Calendar calendar = Calendar.getInstance();
        int year = calendar.get(Calendar.YEAR);
        int monthOfYear = calendar.get(Calendar.MONTH);
        int dayOfMonth = calendar.get(Calendar.DAY_OF_MONTH);
        datePicker.init(year, monthOfYear, dayOfMonth,
                new OnDateChangedListener() {

                    public void onDateChanged(DatePicker view, int year,
                            int monthOfYear, int dayOfMonth) {
                        dateEt.setText("您选择的日期是：" + year + "年"
                                + (monthOfYear + 1) + "月" + dayOfMonth);
                    }
                });
    }
}
```

```
        }

    });

timePicker.setIs24HourView(true); // 设置是否是24小时显示方式
timePicker.setOnTimeChangedListener(new OnTimeChangedListener() {

    public void onTimeChanged(TimePicker view, int hourOfDay, int minute) {
        timeEt.setText("您选择的时间是：" + hourOfDay + "时" + minute +
    }

});

}
```

运行结果:



1.7 Android控件之TextView探究

发表时间: 2011-05-31 关键字: Android, ITeye

在android中，文本控件主要包括TextView控件和EditText控件，本节先对TextView控件的用法进行详细介绍。

TextView类继承自View类，TextView控件的功能是向用户显示文本的内容，但不允许编辑，而其子类EditText允许用户进行编辑。

以下为TextView常用属性及对应方法说明

属性名称	对应方法	说明
android:autoLink	setAutoLinkMask(int)	设置是否将指定格式的文本转换为可单击的超链接显示。 可选的参数值如下： none web: URL链接 email: 邮箱 phone: 电话号码 map: 地图 all
android:gravity	setGravity(int)	设置文本的显示位置
android:height	setHeight(int)	设置文本的高度，单位为px
android:minHeight	setMinHeight(int)	设置文本的最小高度，单位为px
android:maxHeight	setMaxHeight(int)	设置文本的最大高度，单位为px
android:width	setWidth(int)	设置文本的宽度，单位为px
android:minWidth	setMinWidth(int)	设置文本的最小宽度，单位为px
android:maxWidth	setMaxWidth(int)	设置文本的最大宽度，单位为px
android:hint	setHint(int)	当文本为空时显示该文本
android:text	setText(CharSequence)	设置文本内容
android:textColor	setTextColor(ColorStateList)	设置文本颜色
android:textSize	setTextSize(float)	设置文本大小
android:typeface	setTypeface(Typeface)	设置文本字体
android:ellipsize	SetEllipse(TextUtils.TruncateAt)	设置当文字过长时，该控件如何显示。 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)

1.8 Android控件之EditView探究

发表时间: 2011-05-31 关键字: Android, ITeye

EditView类继承自TextView类，EditView与TextView最大的不同就是用户可以对EditView控件进行编辑，同时还可以为EditView控件设置监听器，用来判断用户的输入是否合法。

以下为EditView常用属性及对应方法说明

属性名称	对应方法	说明
android:cursorVisible	setCursorVisible(boolean)	设置光标是否可见，默认可见
android:lines	setLines(int)	设置编辑文本的行数
android:maxLines	setMaxLines(int)	设置编辑文本的最大行数
android:minLines	setMinLines(int)	设置编辑文本的最小行数
android:password	setTransformationMethod (TransformationMethod)	设置文本框中的内容是否显示为密码
android:phoneNumber	setKeyListener(KeyListener)	设置文本中的内容只能是电话号码
android:scrollHorizontally	setHorizontallyScrolling(bool)	是否设置为水平滚动
android:selectAllOnFocus	setSelectAllOnFocus	当文本获得焦点时自动选中全部文本内容
android:singleLine	setTransformationMethod (TransformationMethod)	设置文本为单行模式
android:maxLength	setFilters(InputFilter)	设置最大显示长度

```
/**  
 * 设置当EditText为空，则提示“请输入账号”  
 * 在配置文件main.xml中可以用android:hint="请输入账号"来实现  
 */  
editView.setHint("请输入账号");
```

1.9 Android Activity之间跳转相互传值

发表时间: 2011-05-31 关键字: Android, Java, ITeye, XML, OS

Intent用法实例:

1.无参数Activity跳转

```
Intent it = new Intent(Activity.Main.this, Activity2.class);
startActivity(it);
```

2.向下一个Activity传递数据 (使用Bundle和Intent.putExtra)

```
Intent it = new Intent(Activity.Main.this, Activity2.class);
Bundle bundle=new Bundle();
bundle.putString("name", "This is from MainActivity!");
it.putExtra(bundle); // it.putExtra( "test" , "shuju" );
startActivityForResult(intent, 0);
```

对于数据的获取可以采用 :

```
Bundle bundle = getIntent().getExtras();
String name = bundle.getString("name");
```

AndroidManifest.xml配置文件:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="iaiai.test" android:versionCode="1" android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".MyList" android:label="@string/contact_title">
```

```
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity android:name=".TestWidget" android:label="@string/contact_title"></activity>
</application>
<uses-permission android:name="android.permission.READ_CONTACTS" />
<!-- 在SD卡中创建文件与删除文件权限 -->
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<!-- 往SD卡中写入数据权限 -->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
</manifest>
```

*此文件要注意一个地方就是写的所有Activity类必须要在此配置文件中配置activity，如果不配置则跳转会报错。

main.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <EditText android:id="@+id/et" android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
    <Button android:id="@+id/bt" android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:text="@string/btn_next"></Button>
</LinearLayout>
```

list_items.xml布局文件:

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

```
<EditText android:id="@+id/et1" android:layout_width="fill_parent"
          android:layout_height="wrap_content" />
<Button android:id="@+id/bt1" android:layout_width="fill_parent"
          android:layout_height="wrap_content" android:text="@string/btn_return"></Button>
</LinearLayout>
```

values/strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, TestListView!</string>
    <string name="app_name">TestListView</string>

    <string name="contact_title">Contact</string>

    <string name="btn_next">Next</string>
    <string name="btn_return">Prev</string>
</resources>
```

values-zh-rCN/strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, TestListView!</string>
    <string name="app_name">TestListView</string>

    <string name="contact_title">联系人</string>

    <string name="btn_next">下一步</string>
    <string name="btn_return">返回</string>
</resources>
```

MyList.java类:

```
package iaiai.test;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class MyList extends Activity {

    private EditText editText;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        editText = (EditText) findViewById(R.id.et);

        Button bt = (Button) findViewById(R.id.bt);
        bt.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MyList.this, TestWidget.class);

                Bundle bundle = new Bundle();
                bundle.putString("text", editText.getText().toString());
                intent.putExtras(bundle);

                startActivityForResult(intent, 0);
                // finish(); //这句意思是关闭当前Activity，也就是说点返回键返回不到
            }
        });
    }
}
```

```
/*
 * 取到回传的值
 */
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    switch (resultCode) { // resultCode为回传的标记，我在B中回传的是RESULT_OK
    case RESULT_OK:
        Bundle b = data.getExtras(); // data为B中回传的Intent
        editView.setText(b.getString("text")); // 回传的值
        break;
    default:
        break;
    }
}

}
```

TestWidget.java类：

```
package iaiai.test;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class TestWidget extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.list_items);

        Bundle bundle = getIntent().getExtras();
```

```
final EditText editView = (EditText) findViewById(R.id.et1);
editView.setText(bundle.getString("text"));

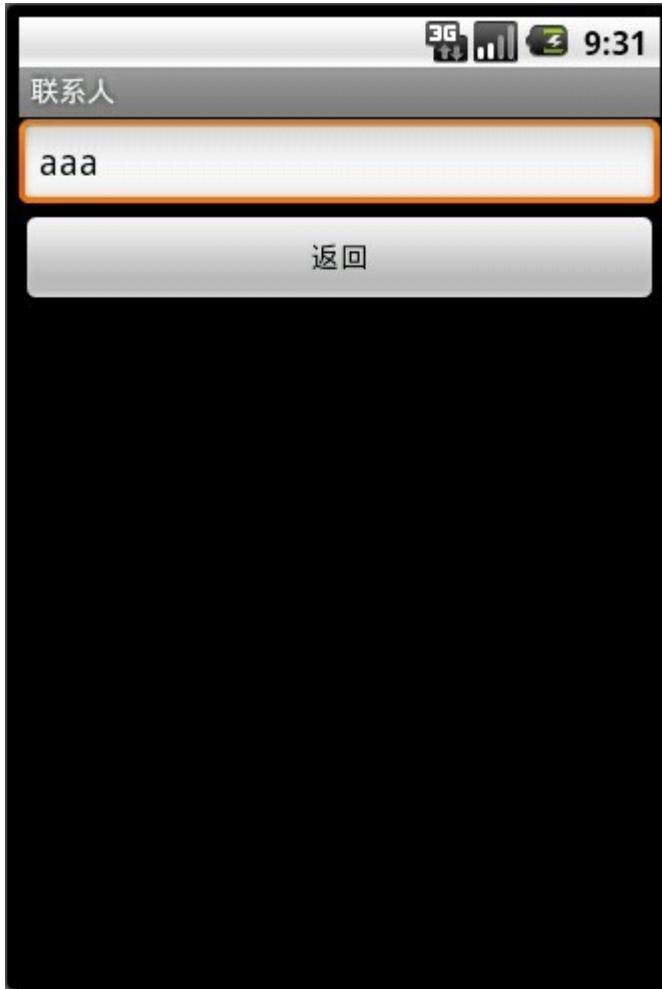
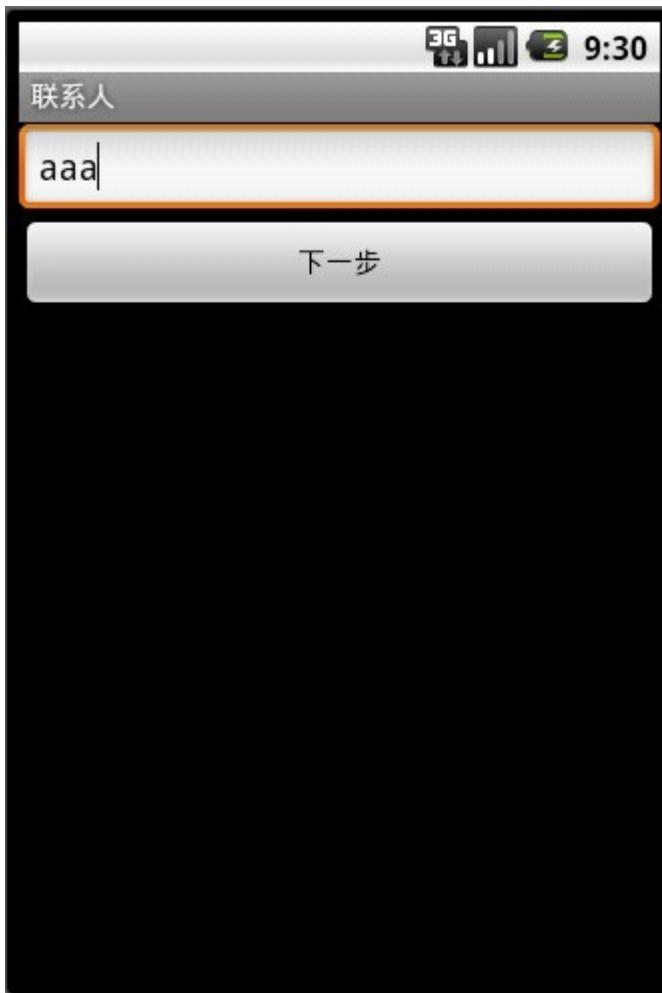
Button bt = (Button) findViewById(R.id.bt1);
bt.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(TestWidget.this, MyList.class);

        Bundle bundle = new Bundle();
        bundle.putString("text", editView.getText().toString());
        intent.putExtras(bundle);

        setResult(RESULT_OK, intent);
        finish(); // 这句意思是关闭当前Activity，也就是说点返回键返回不到这
    }
});
```

}

以下是运行结果：





1.10 Android之Intent探究

发表时间: 2011-05-31 关键字: Android, Google, XML, F#

在一个Android应用中，主要由四种组件组成（四种组件分别为：Activity、Broadcast、Service、ContentProvider），而这四种组件是独立的，它们之间可以互相调用，协调工作，最终组成一个真正的Android应用。在这些组件之间的通讯中，主要是由Intent协助完成的。

Intent负责对应用中一次操作的动作、动作涉及数据、附加数据进行描述，Android则根据此Intent的描述，负责找到对应的组件，将Intent传递给调用的组件，并完成组件的调用。

因此，Intent在这里起着一个媒体中介的作用，专门提供组件互相调用的相关信息，实现调用者与被调用者之间的解耦。

例如，在一个联系人维护的应用中，当我们在一个联系人列表屏幕（假设对应的Activity为ListActivity）上，点击某个联系人后，希望能够跳出此联系人的详细信息屏幕（假设对应的Activity为DetailActivity）为了实现这个目的，ListActivity需要构造一个Intent，这个Intent用于告诉系统，我们要做“查看”动作，此动作对应的查看对象是“某联系人”，然后调用startActivity(Intent intent)，将构造的Intent传入，系统会根据此Intent中的描述，到AndroidManifest.xml中找到满足此Intent要求的Activity，系统会调用找到的Activity，即为DetailActivity，最终传入Intent，DetailActivity则会根据此Intent中的描述，执行相应的操作。

引用

显示网页

```
Uri uri = Uri.parse("http://www.google.com.hk/");  
Intent intent = new Intent(Intent.ACTION_VIEW, uri);  
startActivity(intent);
```

显示地图

```
Uri uri = Uri.parse("geo:38.899533,-77.036476");  
Intent intent = new Intent(Intent.ACTION_VIEW, uri);  
startActivity(intent);  
//其他geo URI范例 ( latitude: 纬度,longitude: 经度 )  
//geo:latitude,longitude  
//geo:latitude,longitude?z=zoom  
//geo:0,0?q=my+street+address  
//geo:0,0?q=business+near+city
```

```
//google.streetview:cbll=lat,lng&cbp=1,yaw,,pitch,zoom&mz=mapZoom
```

路径规划

```
Uri uri = Uri.parse("http://maps.google.com/
maps?f=d&saddr=startLat%20startLng&daddr=endLat%20endLng&hl=en");
Intent intent = new Intent(Intent.ACTION_VIEW, uri);
startActivity(intent);
//where startLat, startLng, endLat, endLng are a long with 6 decimals like: 50.123456
```

打电话

```
//叫出拨号程序
Uri uri = Uri.parse("tel:0800000123");
Intent intent = new Intent(Intent.ACTION_DIAL, uri);
startActivity(intent);
```

//直接打电话出去

```
Uri uri = Uri.parse("tel:0800000123");
Intent intent = new Intent(Intent.ACTION_CALL, uri);
startActivity(intent);
//用這個，要在 AndroidManifest.xml 中，加上<uses-permission
id="android.permission.CALL_PHONE" />
```

传送SMS/MMS

```
//调用短信程序
Intent it = new Intent(Intent.ACTION_VIEW, uri);
it.putExtra("sms_body", "The SMS text");
it.setType("vnd.android-dir/mms-sms");
startActivity(it);
```

//传送消息

```
Uri uri = Uri.parse("smsto://0800000123");
Intent it = new Intent(Intent.ACTION_SENDTO, uri);
it.putExtra("sms_body", "The SMS text");
startActivity(it);
```

//传送 MMS

```
Uri uri = Uri.parse("content://media/external/images/media/23");
Intent it = new Intent(Intent.ACTION_SEND);
it.putExtra("sms_body", "some text");
it.putExtra(Intent.EXTRA_STREAM, uri);
it.setType("image/png");
startActivity(it);
```

传送 Email

```
Uri uri = Uri.parse("mailto:xxx@abc.com");
Intent it = new Intent(Intent.ACTION_SENDTO, uri);
startActivity(it);

Intent it = new Intent(Intent.ACTION_SEND);
it.putExtra(Intent.EXTRA_EMAIL, "me@abc.com");
it.putExtra(Intent.EXTRA_TEXT, "The email body text");
it.setType("text/plain");
startActivity(Intent.createChooser(it, "Choose Email Client"));
```

```
Intent it=new Intent(Intent.ACTION_SEND);
```

```
String[] tos={"me@abc.com"};
String[] ccs={"you@abc.com"};
it.putExtra(Intent.EXTRA_EMAIL, tos);
it.putExtra(Intent.EXTRA_CC, ccs);
it.putExtra(Intent.EXTRA_TEXT, "The email body text");
it.putExtra(Intent.EXTRA_SUBJECT, "The email subject text");
it.setType("message/rfc822");
startActivity(Intent.createChooser(it, "Choose Email Client"));
```

//传送附件

```
Intent it = new Intent(Intent.ACTION_SEND);
it.putExtra(Intent.EXTRA_SUBJECT, "The email subject text");
it.putExtra(Intent.EXTRA_STREAM, "file:///sdcard/mysong.mp3");
sendIntent.setType("audio/mp3");
startActivity(Intent.createChooser(it, "Choose Email Client"));
```

播放多媒体

```
Uri uri = Uri.parse("file:///sdcard/song.mp3");
Intent it = new Intent(Intent.ACTION_VIEW, uri);
it.setType("audio/mp3");
startActivity(it);
```

```
Uri uri = Uri.withAppendedPath(MediaStore.Audio.Media.INTERNAL_CONTENT_URI, "1");
Intent it = new Intent(Intent.ACTION_VIEW, uri);
startActivity(it);
```

Market 相关

```
//寻找某个应用
Uri uri = Uri.parse("market://search?q=pname:pkg_name");
Intent it = new Intent(Intent.ACTION_VIEW, uri);
startActivity(it);
//where pkg_name is the full package path for an application
```

```
//显示某个应用的相关信息
```

```
Uri uri = Uri.parse("market://details?id=app_id");
Intent it = new Intent(Intent.ACTION_VIEW, uri);
startActivity(it);
//where app_id is the application ID, find the ID
//by clicking on your application on Market home
//page, and notice the ID from the address bar
```

Uninstall 应用程序

```
Uri uri = Uri.fromParts("package", strPackageName, null);
Intent it = new Intent(Intent.ACTION_DELETE, uri);
startActivity(it);
```

1.11 Android 电话拨号

发表时间: 2011-05-31 关键字: Android, ITeye, Java, XML, OS

以下案例模拟android电话拨号器的实现

AndroidManifest.xml清单列表:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ljq.phone"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".MyList" android:label="@string/contact_title">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="7" />
    <uses-permission android:name="android.permission.CALL_PHONE"/>
</manifest>
```

main.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="请输入电话号码" />
    <EditText android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/phone" />
```

```
<Button android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="拨打此号码"
    android:id="@+id/button" />
</LinearLayout>
```

MyList.java类:

```
package iaiai.test;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MyList extends Activity {

    private EditText phone = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        phone = (EditText) this.findViewById(R.id.phone);
        Button button = (Button) this.findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                String tel = phone.getText().toString();
                // 方法一， 使用Intent目的： 激活android组件
                // Intent intent=new Intent();
                // intent.setAction("android.intent.action.CALL");
                // intent.setData(Uri.parse("tel:"+tel));
            }
        });
    }
}
```

```
// 方法二
Intent intent = new Intent("android.intent.action.CALL", Uri
    .parse("tel:" + tel));
// 方法的内部会自动为intent对象设置类别:android.intent.category.D
startActivity(intent);
}

});

}

}
```

运行结果：





1.12 Android 短信发送器

发表时间: 2011-05-31 关键字: Android, Mobile, Java, ITeye, XML

模拟android短信发送器的实现

AndroidManifest.xml清单文件:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ljq.sms"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".MainActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="7" />
    <uses-permission android:name="android.permission.SEND_SMS"/>
</manifest>
```

main.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <RelativeLayout android:layout_width="fill_parent"
        android:layout_height="wrap_content">
        <TextView android:layout_width="115dip"
            android:layout_height="wrap_content" />
    </RelativeLayout>
</LinearLayout>
```

```
        android:layout_height="wrap_content"
        android:text="请输入手机号"
        android:id="@+id/mobilelabel" />
<EditText android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_toRightOf="@+id/mobilelabel"
        android:text="5556"
        android:id="@+id/mobile" />
</RelativeLayout>
<TextView android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="请输入短信内容" />
<EditText android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:minLines="3"
        android:text="I am a teacher!"
        android:id="@+id/content" />
<Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="发送"
        android:id="@+id/button" />
</LinearLayout>
```

MainActivity.java类：

```
package com.ljq.sms;

import java.util.ArrayList;

import android.app.Activity;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

```
public class MainActivity extends Activity {  
    private EditText mobileText=null;  
    private EditText contentText=null;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
  
        mobileText=(EditText)findViewById(R.id.mobile);  
        contentText=(EditText)findViewById(R.id.content);  
        Button button=(Button)findViewById(R.id.button);  
        button.setOnClickListener(new View.OnClickListener(){  
  
            public void onClick(View v) {  
                String mobile=mobileText.getText().toString();  
                String content=contentText.getText().toString();  
  
                //取得android系统中默认的短信管理器  
                SmsManager manager=SmsManager.getDefault();  
                //如果短信内容过长时，则对短信内容进行拆分  
                ArrayList<String> texts=manager.divideMessage(content);  
                for(String text:texts){  
                    //第一个参数：对方手机号码  
                    //第二个参数：短信中心号码，一般设置为空  
                    //第三个参数：短信内容  
                    //第四个参数：sentIntent判断短信是否发送成功，如果你没有SIM卡，或者网络中断，则可  
                    //注意强调的是“发送”的动作是否成功。那么至于对于对方是否收到，另当别论  
                    //第五个参数：当短信发送到收件人时，会收到这个deliveryIntent。即强调了“发送”后  
                    //就是说是在“短信发送成功”和“对方收到此短信”才会激活sentIntent和deliveryIntent  
                    manager.sendTextMessage(mobile, null, text, null, null);  
                }  
                //Toast.makeText(getApplicationContext(), "发送成功", Toast.LENGTH_LONG).show()  
                Toast.makeText(MainActivity.this, "发送成功", Toast.LENGTH_LONG).show();  
            }  
        });  
    }  
}
```



1.13 Android 中的显示单位

发表时间: 2011-05-31 关键字: Android

px (pixels)像素

一般HVGA代表320x480像素，这个用的比较多。

dip或dp (device independent pixels)设备独立像素

这个和设备硬件有关，一般为了支持WVGA、HVGA和QVGA 推荐使用这个，不依赖像素。

sp (scaled pixels — best for text size)比例像素

主要处理字体的大小，可以根据系统的字体自适应。

除了上面三个显示单位，下面还有几个不太常用：

in (inches)英寸

mm (millimeters)毫米

pt (points)点，1/72英寸

为了适应不同分辨率，不同的像素密度，推荐使用dip，文字使用sp。

1.14 Android 使用文件进行数据存储

发表时间: 2011-05-31 关键字: Android, 网络应用, Myeclipse, SQLite, J2SE

很多时候我们开发的软件需要对处理后的数据进行存储,以供再次访问。Android为数据存储提供了如下几种方式：

- 文件
- SharedPreferences(参数)
- SQLite数据库
- 内容提供者 (Content provider)
- 网络

首先给大家介绍使用文件如何对数据进行存储

Activity提供了openFileOutput()方法可以用于把数据输出到文件中，具体的实现过程与在J2SE环境中保存数据到文件中是一样的

```
public class FileActivity extends Activity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        FileOutputStream outStream = this.openFileOutput("ljq.txt", Context.MODE_PRIVATE);  
        outStream.write("测试数据...".getBytes());  
        outStream.close();  
    }  
}
```

openFileOutput(fileName, mode)方法详解:

第一参数：

用于指定文件名称，不能包含路径分隔符 “/” 。如果文件不存在，Android 会自动创建它。创建的文件保存在 /data/data/<packagename>/files目录，如：/data/data/com.ljq.activity/files/itcast.txt，通过点击 MyEclipse菜单 “Window” - “Show View” - “Other” ，在对话窗口中展开android文件夹，选择下面的File Explorer视图，然后在File Explorer视图中展开/data/<package name>/files目录就可以看到该文件

第二参数

用于指定操作模式，有四种模式

Context.MODE_PRIVATE = 0

默认操作模式，代表该文件是私有数据，只能被应用本身访问，在该模式下，写入的内容会覆盖原文件的内容，如果想把新写入的内容追加到原文件中。

可以使用Context.MODE_APPEND。

Context.MODE_APPEND = 32768

检查文件是否存在，存在就往文件追加内容，否则就创建新文件。

Context.MODE_WORLD_READABLE = 1

表示当前文件可以被其他应用读取

Context.MODE_WORLD_WRITEABLE = 2

表示当前文件可以被其他应用写入

注意：

Context.MODE_WORLD_READABLE和Context.MODE_WORLD_WRITEABLE用来控制其他应用是否有权限读写该文件。

如果希望文件被其他应用读和写，可以传入：openFileOutput("ljq.txt",

Context.MODE_WORLD_READABLE + Context.MODE_WORLD_WRITEABLE);

android有一套自己的安全模型，当应用程序(.apk)在安装时系统就会分配给他一个userid，当该应用要去访问其他资源比如文件的时候，就需要userid匹配。默认情况下，任何应用创建的文件，sharedpreferences，数据库都应该是私有的（位于/data/data/<package name>/files），其他程序无法访问。除非在创建时指定了Context.MODE_WORLD_READABLE或者Context.MODE_WORLD_WRITEABLE，只有这样其他程序才能正确访问。

如果要打开存放在/data/data/<package name>/files目录应用私有的文件，可以使用Activity提供openFileInput()方法。

```
FileInputStream inStream = this.getContext().openFileInput("ljq.txt");
```

或者直接使用文件的绝对路径：

```
File file = new File("/data/data/com.ljq.action/files/ljq.txt");
FileInputStream inStream = new FileInputStream(file);
```

注意：上面文件路径中的“com.ljq.action”为应用所在包，当你在编写代码时应替换为你自己应用使用的包。对于私有文件只能被创建该文件的应用访问，如果希望文件能被其他应用读和写，可以在创建文件时，指定Context.MODE_WORLD_READABLE和Context.MODE_WORLD_WRITEABLE权限。

Activity还提供了getCacheDir()和getFilesDir()方法：

getCacheDir()方法用于获取/data/data/<package name>/cache目录

getFilesDir()方法用于获取/data/data/<package name>/files目录

案例：

FileUtil.java类：

```
package iaiai.test;

import java.io.ByteArrayOutputStream;
import java.io.InputStream;
import java.io.OutputStream;

public class FileUtil {

    /**
     * 保存数据
     *
     * @param outputStream
     * @param content
     * @throws Exception
     */
    public static void save(OutputStream outputStream, String content)
            throws Exception {
        outputStream.write(content.getBytes());
        outputStream.close();
    }

    /**
     * 读取数据
     *
     * @param inputStream
     * @return
     * @throws Exception
     */
    public static String read(InputStream inputStream) throws Exception {
        // 往内存写数据
        ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
    }
}
```

```
// 缓冲区
byte[] buffer = new byte[1024];
int len = -1;
while ((len = inputStream.read(buffer)) != -1) {
    byteArrayOutputStream.write(buffer, 0, len);
}
// 存储数据
byte[] data = byteArrayOutputStream.toByteArray();
byteArrayOutputStream.close();
inputStream.close();
return new String(data);
}

}
```

MyList.java类:

```
package iaiai.test;

import android.app.Activity;
import android.content.Context;
import android.os.Bundle;

public class MyList extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        try {
            FileUtil.save(openFileOutput("test.txt", Context.MODE_PRIVATE),
                    "这是测试数据.....");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

}

1.15 Android 把文件存放在SDCard

发表时间: 2011-05-31 关键字: Android, Java, DOS, OS, Eclipse

使用Activity的openFileOutput()方法保存文件，文件是存放在手机空间上，一般手机的存储空间不是很大，存放些小文件还行，如果要存放像视频这样的大文件，是不可行的。对于像视频这样的大文件，我们可以把它存放在SDCard。 SDCard是干什么的？你可以把它看作是移动硬盘或U盘。

在模拟器中使用SDCard，你需要先创建一张SDCard卡（当然不是真的SDCard，只是镜像文件）。创建SDCard可以在Eclipse创建模拟器时随同创建，也可以使用DOS命令进行创建，如下：在Dos窗口中进入android SDK安装路径的tools目录，输入以下命令创建一张容量为2G的SDCard，文件后缀可以随便取，建议使用.img：

引用

```
mksdcard 2048M D:\AndroidTool\sdcard.img
```

注意：在程序中访问SDCard，你需要申请访问SDCard的权限。

在AndroidManifest.xml中加入访问SDCard的权限如下：

```
<!-- 在SDCard中创建与删除文件权限 -->
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
<!-- 往SDCard写入数据权限 -->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

要往SDCard存放文件，程序必须先判断手机是否装有SDCard，并且可以进行读写。

注意：访问SDCard必须在AndroidManifest.xml中加入访问SDCard的权限

```
if(Environment.getExternalStorageState().equals(Environment.MEDIA_MOUNTED)){
    File sdCardDir = Environment.getExternalStorageDirectory();//获取SDCard目录
    File saveFile = new File(sdCardDir, "text.txt");
    FileOutputStream outStream = new FileOutputStream(saveFile);
    outStream.write("测试数据.....".getBytes());
    outStream.close();
}
```

Environment.getExternalStorageState()方法用于获取SDCard的状态，如果手机装有SDCard，并且可以进行读写，

那么方法返回的状态等于Environment.MEDIA_MOUNTED。

Environment.getExternalStorageDirectory()方法用于获取SDCard的目录，当然要获取SDCard的目录，你也可以这样写：

```
File saveFile = new File("/mnt/sdcard/text.txt");
FileOutputStream outStream = new FileOutputStream(saveFile);
outStream.write("abc".getBytes());
outStream.close();
```

案例：

MyList.java类：

```
package iaiai.test;

import java.io.File;
import java.io.FileOutputStream;

import android.app.Activity;
import android.os.Bundle;
import android.os.Environment;
import android.widget.Toast;

public class MyList extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        try {
            // sdcard存在并且可以读写
            if (Environment.getExternalStorageState().equals(
                    Environment.MEDIA_MOUNTED)) {
                File sdCardDir = Environment.getExternalStorageDirectory(); //
```

```
        File saveFile = new File(sdCardDir, "test.txt");

        FileUtil.save(new FileOutputStream(saveFile), "这是测试数据.....
    } else {
        Toast.makeText(MyList.this, "sdcard不存在或写保护", Toast.LENGTH
            .show();
    }
} catch (Exception e) {
    e.printStackTrace();
}
}
}
```

FileUtil.java类:

```
package iaiai.test;

import java.io.ByteArrayOutputStream;
import java.io.InputStream;
import java.io.OutputStream;

public class FileUtil {

    /**
     * 保存数据
     *
     * @param outputStream
     * @param content
     * @throws Exception
     */
    public static void save(OutputStream outputStream, String content)
        throws Exception {
        outputStream.write(content.getBytes());
        outputStream.close();
    }
}
```

```
}

/**
 * 读取数据
 *
 * @param inputStream
 * @return
 * @throws Exception
 */

public static String read(InputStream inputStream) throws Exception {
    // 往内存写数据
    ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
    // 缓冲区
    byte[] buffer = new byte[1024];
    int len = -1;
    while ((len = inputStream.read(buffer)) != -1) {
        byteArrayOutputStream.write(buffer, 0, len);
    }
    // 存储数据
    byte[] data = byteArrayOutputStream.toByteArray();
    byteArrayOutputStream.close();
    inputStream.close();
    return new String(data);
}

}
```

1.16 Android SQLite数据库增删改查操作

发表时间: 2011-05-31 关键字: SQLite, Android, 数据结构, SQL, 嵌入式

一、使用嵌入式关系型SQLite数据库存储数据

在Android平台上，集成了一个嵌入式关系型数据库——SQLite，SQLite3支持NULL、INTEGER、REAL（浮点数字）、TEXT（字符串文本）和BLOB（二进制对象）数据类型，虽然它支持的类型只有五种，但实际上sqlite3也接受varchar(n)、char(n)、decimal(p,s)等数据类型，只不过在运算或保存时会转成对应的五种数据类型。

SQLite最大的特点是你可以把各种类型的数据保存到任何字段中，而不用关心字段声明的数据类型是什么。例如：可以在Integer类型的字段中存放字符串，或者在布尔型字段中存放浮点数，或者在字符型字段中存放日期型值。但有一种情况例外：定义为INTEGER PRIMARY KEY的字段只能存储64位整数，当向这种字段保存除整数以外的数据时，将会产生错误。另外，在编写CREATE TABLE语句时，你可以省略跟在字段名称后面的数据类型信息，如下面语句你可以省略name字段的类型信息：

```
CREATE TABLE person (personid integer primary key autoincrement, name varchar(20))
```

SQLite可以解析大部分标准SQL语句，如：

引用

查询语句：select * from 表名 where 条件子句 group by 分组字句 having ... order by 排序子句

如：

```
select * from person
```

```
select * from person order by id desc
```

```
select name from person group by name having count(*)>1
```

分页SQL与mysql类似，下面SQL语句获取5条记录，跳过前面3条记录

```
select * from Account limit 5 offset 3 或者 select * from Account limit 3,5
```

插入语句：insert into 表名(字段列表) values(值列表)。如：insert into person(name, age) values(‘丸子’ ,3)

更新语句：update 表名 set 字段名=值 where 条件子句。如：update person set name= ‘丸子 ’ where id=10

删除语句：delete from 表名 where 条件子句。如：delete from person where id=10

二、使用SQLiteOpenHelper对数据库进行版本管理

我们在编写数据库应用软件时，需要考虑这样的问题：因为我们开发的软件可能会安装在很多用户的手机上，如果应用使用到了SQLite数据库，我们必须在用户初次使用软件时创建出应用使用的数据库表结构及添加一些初始化记录，另外在软件升级的时候，也需要对数据表结构进行更新。那么，我们如何才能实现在用户初次

使用或升级软件时自动在用户的手机上创建出应用需要的数据库表呢？总不能让我们在每个需要安装此软件的手机上通过手工方式创建数据库表吧？因为这种需求是每个数据库应用都要面临的，所以在Android系统，为我们提供了一个名为SQLiteOpenHelper的抽象类，必须继承它才能使用，它是通过对数据库版本进行管理来实现前面提出的需求。

为了实现对数据库版本进行管理，SQLiteOpenHelper类提供了两个重要的方法，分别是onCreate(SQLiteDatabase db)和onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)，前者用于初次使用软件时生成数据库表，后者用于升级软件时更新数据库表结构。当调用SQLiteOpenHelper的getWritableDatabase()或者getReadableDatabase()方法获取用于操作数据库的SQLiteDatabase实例的时候，如果数据库不存在，Android系统会自动生成一个数据库，接着调用onCreate()方法，onCreate()方法在初次生成数据库时才会被调用，在onCreate()方法里可以生成数据库表结构及添加一些应用使用到的初始化数据。onUpgrade()方法在数据库的版本发生变化时会被调用，一般在软件升级时才需改变版本号，而数据库的版本是由程序员控制的，假设数据库现在的版本是1，由于业务的变更，修改了数据库表结构，这时候就需要升级软件，升级软件时希望更新用户手机里的数据库表结构，为了实现这一目的，可以把原来的数据库版本设置为2(有同学问设置为3行不行？当然可以，如果你愿意，设置为100也行)，并且在onUpgrade()方法里面实现表结构的更新。当软件的版本升级次数比较多，这时在onUpgrade()方法里面可以根据原版号和目标版本号进行判断，然后作出相应的表结构及数据更新。

getWritableDatabase()和getReadableDatabase()方法都可以获取一个用于操作数据库的SQLiteDatabase实例。但getWritableDatabase()方法以读写方式打开数据库，一旦数据库的磁盘空间满了，数据库就只能读而不能写，倘若使用getWritableDatabase()打开数据库就会出错。getReadableDatabase()方法先以读写方式打开数据库，如果数据库的磁盘空间满了，就会打开失败，当打开失败后会继续尝试以只读方式打开数据库。注意：getWritableDatabase()，getReadableDatabase的区别是当数据库写满时，调用前者会报错，调用后者不会，所以如果不是更新数据库的话，最好调用后者来获得数据库连接。

代码：

```
public class DatabaseHelper extends SQLiteOpenHelper {  
    //类没有实例化，是不能用作父类构造器的参数，必须声明为静态  
    private static final String name = "ljqdb"; //数据库名称  
    private static final int version = 1; //数据库版本  
    public DatabaseHelper(Context context) {  
        //第三个参数CursorFactory指定在执行查询时获得一个游标实例的工厂类，设置为null，代表使用系统默认  
        super(context, name, null, version);  
    }  
  
    @Override  
    public void onCreate(SQLiteDatabase db) {  
        db.execSQL("CREATE TABLE IF NOT EXISTS person (
```

```
    personid integer primary key autoincrement, name varchar(20), age INTEGER));  
}  
  
@Override  
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
    db.execSQL(" ALTER TABLE person ADD phone VARCHAR(12) NULL "); //往表中增加一列  
    // DROP TABLE IF EXISTS person 删除表  
}  
}
```

在实际项目开发中，当数据库表结构发生更新时，应该避免用户存放于数据库中的数据丢失。

三、使用SQLiteDatabase操作SQLite数据库

Android提供了一个名为SQLiteDatabase的类，该类封装了一些操作数据库的API，使用该类可以完成对数据进行添加(Create)、查询(Retrieve)、更新(Update)和删除(Delete)操作（这些操作简称为CRUD）。对SQLiteDatabase的学习，我们应该重点掌握execSQL()和rawQuery()方法。execSQL()方法可以执行insert、delete、update和CREATE TABLE之类有更改行为的SQL语句；rawQuery()方法用于执行select语句。execSQL()方法的使用例子：

```
SQLiteDatabase db = ....;  
db.execSQL("insert into person(name, age) values('丸子', 24)");  
db.close();
```

执行上面SQL语句会往person表中添加进一条记录，在实际应用中，语句中的“丸子”这些参数值会由用户输入界面提供，如果把用户输入的内容原样组拼到上面的insert语句，当用户输入的内容含有单引号时，组拼出来的SQL语句就会存在语法错误。要解决这个问题需要对单引号进行转义，也就是把单引号转换成两个单引号。有些时候用户往往还会输入像“&”这些特殊SQL符号，为保证组拼好的SQL语句语法正确，必须对SQL语句中的这些特殊SQL符号都进行转义，显然，对每条SQL语句都做这样的处理工作是比较烦琐的。

SQLiteDatabase类提供了一个重载后的execSQL(String sql, Object[] bindArgs)方法，使用这个方法可以解决前面提到的问题，因为这个方法支持使用占位符参数(?)。使用例子如下：

```
SQLiteDatabase db = ....;  
db.execSQL("insert into person(name, age) values(?,?)", new Object[]{"丸子", 4});  
db.close();
```

execSQL(String sql, Object[] bindArgs)方法的第一个参数为SQL语句，第二个参数为SQL语句中占位符参数的值，参数值在数组中的顺序要和占位符的位置对应。

SQLiteDatabase的rawQuery()用于执行select语句，使用例子如下：

```
SQLiteDatabase db = ....;
Cursor cursor = db.rawQuery("select * from person", null);
while (cursor.moveToNext()) {
    int personid = cursor.getInt(0); //获取第一列的值,第一列的索引从0开始
    String name = cursor.getString(1); //获取第二列的值
    int age = cursor.getInt(2); //获取第三列的值
}
cursor.close();
db.close();
```

rawQuery()方法的第一个参数为select语句；第二个参数为select语句中占位符参数的值，如果select语句没有使用占位符，该参数可以设置为null。带占位符参数的select语句使用例子如下：

```
Cursor cursor = db.rawQuery("select * from person where name like ? and age=? ", new String[]{"%"});
```

Cursor是结果集游标，用于对结果集进行随机访问，如果大家熟悉jdbc，其实Cursor与JDBC中的ResultSet作用很相似。使用moveToNext()方法可以将游标从当前行移动到下一行，如果已经移过了结果集的最后一行，返回结果为false，否则为true。另外Cursor还有常用的moveToPrevious()方法（用于将游标从当前行移动到上一行，如果已经移过了结果集的第一行，返回值为false，否则为true）、moveToFirst()方法（用于将游标移动到结果集的第一行，如果结果集为空，返回值为false，否则为true）和moveToLast()方法（用于将游标移动到结果集的最后一行，如果结果集为空，返回值为false，否则为true）。

除了前面给大家介绍的execSQL()和rawQuery()方法，SQLiteDatabase还专门提供了对应于添加、删除、更新、查询的操作方法：insert()、delete()、update()和query()。这些方法实际上是给那些不太了解SQL语法的菜鸟使用的，对于熟悉SQL语法的程序员而言，直接使用execSQL()和rawQuery()方法执行SQL语句就能完成数据的添加、删除、更新、查询操作。

Insert()方法用于添加数据，各个字段的数据使用ContentValues进行存放。ContentValues类似于MAP，相对于MAP，它提供了存取数据对应的put(String key, Xxx value)和getAsXxx(String key)方法，key为字段名称，value为字段值，Xxx指的是各种常用的数据类型，如：String、Integer等。

```
SQLiteDatabase db = databaseHelper.getWritableDatabase();
ContentValues values = new ContentValues();
values.put("name", "丸子");
values.put("age", 24);
long rowid = db.insert("person", null, values); //返回新添记录的行号，与主键id无关
```

不管第三个参数是否包含数据，执行Insert()方法必然会添加一条记录，如果第三个参数为空，会添加一条除主键之外其他字段值为Null的记录。Insert()方法内部实际上通过构造insert SQL语句完成数据的添加，Insert()方法的第二个参数用于指定空值字段的名称，相信大家对该参数会感到疑惑，该参数的作用是什么？是这样的：如果第三个参数values 为Null或者元素个数为0，由于Insert()方法要求必须添加一条除了主键之外其它字段为Null值的记录，为了满足SQL语法的需要，insert语句必须给定一个字段名，如：insert into person(name) values(NULL)，倘若不给定字段名，insert语句就成了这样：insert into person() values()，显然这不满足标准SQL的语法。对于字段名，建议使用主键之外的字段，如果使用了INTEGER类型的主键字段，执行类似insert into person(personid) values(NULL)的insert语句后，该主键字段值也不会为NULL。如果第三个参数values 不为Null并且元素的个数大于0，可以把第二个参数设置为null。

delete()方法的使用：

```
SQLiteDatabase db = databaseHelper.getWritableDatabase();
db.delete("person", "personid<?", new String[]{"2"});
db.close();
```

上面代码用于从person表中删除personid小于2的记录。

update()方法的使用：

```
SQLiteDatabase db = databaseHelper.getWritableDatabase();
ContentValues values = new ContentValues();
values.put("name", "丸子"); //key为字段名，value为值
db.update("person", values, "personid=?", new String[]{"1"});
db.close();
```

上面代码用于把person表中personid等于1的记录的name字段的值改为“丸子”。

query()方法实际上是把select语句拆分成了若干个组成部分，然后作为方法的输入参数：

```
SQLiteDatabase db = databaseHelper.getWritableDatabase();

Cursor cursor = db.query("person", new String[]{"personid,name,age"}, "name like ?", new String[]{name});
while (cursor.moveToNext()) {
    int personid = cursor.getInt(0); //获取第一列的值,第一列的索引从0开始
    String name = cursor.getString(1); //获取第二列的值
    int age = cursor.getInt(2); //获取第三列的值
}
cursor.close();
db.close();
```

上面代码用于从person表中查找name字段含有“传智”的记录，匹配的记录按personid降序排序，对排序后的结果略过第一条记录，只获取2条记录。

query(table, columns, selection, selectionArgs, groupBy, having, orderBy, limit)方法各参数的含义：

- table：表名。相当于select语句from关键字后面的部分。如果是多表联合查询，可以用逗号将两个表名分开。
- columns：要查询出来的列名。相当于select语句select关键字后面的部分。
- selection：查询条件子句，相当于select语句where关键字后面的部分，在条件子句允许使用占位符“?”
- selectionArgs：对应于selection语句中占位符的值，值在数组中的位置与占位符在语句中的位置必须一致，否则就会有异常。
- groupBy：相当于select语句group by关键字后面的部分
- having：相当于select语句having关键字后面的部分
- orderBy：相当于select语句order by关键字后面的部分，如：personid desc, age asc;
- limit：指定偏移量和获取的记录数，相当于select语句limit关键字后面的部分。

四、使用SQLiteOpenHelper获取用于操作数据库的SQLiteDatabase实例

```
public class DatabaseHelper extends SQLiteOpenHelper {
    private static final String name = "ljqdb"; //数据库名称
    private static final int version = 1; //数据库版本
    .....
}

public class HelloActivity extends Activity {
    @Override public void onCreate(Bundle savedInstanceState) {
        .....
    }
}
```

```
Button button =(Button) this.findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener(){
public void onClick(View v) {
    DatabaseHelper databaseHelper = new DatabaseHelper(HelloActivity.this);
    SQLiteDatabase db = databaseHelper.getWritableDatabase();
    db.execSQL("insert into person(name, age) values(?,?)", new Object[]{"丸子", 4});
    db.close();
}}
}
```

第一次调用getWritableDatabase()或getReadableDatabase()方法后，SQLiteOpenHelper会缓存当前的 SQLiteDatabase实例，SQLiteDatabase实例正常情况下会维持数据库的打开状态，所以在你不再需要 SQLiteDatabase实例时，请及时调用close()方法释放资源。一旦SQLiteDatabase实例被缓存，多次调用getWritableDatabase()或getReadableDatabase()方法得到的都是同一实例。

五、使用事务操作SQLite数据库

使用SQLiteDatabase的beginTransaction()方法可以开启一个事务，程序执行到endTransaction()方法时会检查事务的标志是否为成功，如果程序执行到endTransaction()之前调用了setTransactionSuccessful()方法设置事务的标志为成功则提交事务，如果没有调用setTransactionSuccessful()方法则回滚事务。

使用例子如下：

```
SQLiteDatabase db = ....;
db.beginTransaction();//开始事务
try {
    db.execSQL("insert into person(name, age) values(?,?)", new Object[]{"丸子", 4});
    db.execSQL("update person set name=? where personid=?", new Object[]{"abc", 1});
    db.setTransactionSuccessful();//调用此方法会在执行到endTransaction()时提交当前事务，如果不调用
} finally {
    db.endTransaction();//由事务的标志决定是提交事务，还是回滚事务
}
db.close();
```

上面两条SQL语句在同一个事务中执行。

1.17 Android 获取联系人

发表时间: 2011-06-01 关键字: Android, Java, ITeye, XML, OS

在Android中，可以使用ContentResolver对通信录中的数据进行添加、删除、修改和查询操作。

在对联系人进行操作时需加入以下两个权限

```
<!-- 添加操作联系人的权限 -->
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
```

引用

```
<!-- 联系人相关的uri -->
content://com.android.contacts/contacts 操作的数据是联系人信息Uri
content://com.android.contacts/data/phones 联系人电话Uri
content://com.android.contacts/data/emails 联系人Email Uri
```

main.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <ListView android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:id="@+id/MyListView">
    </ListView>
</LinearLayout>
```

list_items.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:id="@+id/myListItem"
    android:paddingBottom="5dip" android:paddingLeft="12dip"
```

```
    android:paddingRight="12dip">
    <ImageView android:id="@+id/img" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_alignParentLeft="true"></Im
<TextView android:layout_height="wrap_content"
    android:layout_width="wrap_content" android:id="@+id/itemTitle"
    android:textSize="20dip" android:paddingTop="15px"
    android:layout_marginLeft="70px">
</TextView>
<CheckBox android:id="@+id/cb" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:layout_alignParentRight="true"
    android:focusable="false"></CheckBox>
</RelativeLayout>
```

MyList.java类:

```
package iaiai.test;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.ContentResolver;
import android.content.DialogInterface;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.view.ContextMenu;
import android.view.ContextMenu.ContextMenuInfo;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnCreateContextMenuListener;
import android.widget.AdapterView;
```

```
import android.widget.AdapterView.AdapterContextMenuInfo;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;
import android.widget.Toast;

public class MyList extends Activity {

    private ListView list;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        list = (ListView) findViewById(R.id.MyListView);

        // 配置适配器
        MyCheckBoxAdapter adapter = new MyCheckBoxAdapter(this, getData()); // 布局里的
        // 添加并且显示
        list.setAdapter(adapter);

        // 添加点击
        list.setOnItemClickListener(new OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
                    long arg3) {
                setTitle("点击第" + arg2 + "个项目");
                Toast.makeText(MyList.this, "点击第" + arg2 + "个项目",
                        Toast.LENGTH_LONG).show();
            }
        });

        // 添加长按点击
        list.setOnCreateContextMenuListener(new OnCreateContextMenuListener() {
            @Override
            public void onCreateContextMenu(ContextMenu menu, View v,
                    ContextMenuInfo menuInfo) {
```

```
        // menu.setHeaderIcon(R.drawable.icon); //字义图片样式
        menu.setHeaderTitle("长按菜单选项:");
        menu.add(1, 0, 0, "编辑");
        menu.add(0, 1, 0, "删除");
    }
});
```

```
}
```

```
@Override
public boolean onContextItemSelected(MenuItem item) {
    int selectedPosition = ((AdapterContextMenuInfo) item.getMenuInfo()).position;
    Map<String, String> map = (HashMap) list
        .getItemAtPosition(selectedPosition); // ListView中的数据
    Toast.makeText(MyList.this,
        "你选择的:" + selectedPosition + "，内容为:" + map.get("itemTitl
        Toast.LENGTH_LONG).show(); // 显示那条数据

    String temp = "";
    switch (item.getItemId()) {
        case 0:
            temp = "编辑";
            break;
        case 1:
            temp = "删除";
            break;
        default:
            break;
    }
    // Toast.makeText(this, temp + "處理", Toast.LENGTH_LONG).show();
    return super.onContextItemSelected(item);
}

/**
 * 获取联系人
 *
 * @return
 */
```

```
private List<Map<String, String>> getData() {  
    // 组织数据源  
    List<Map<String, String>> mylist = new ArrayList<Map<String, String>>();  
  
    ContentResolver contentResolver = getContentResolver();  
    Uri uri = Uri.parse("content://com.android.contacts/contacts");  
    Cursor cursor = contentResolver.query(uri, null, null, null, null);  
    while (cursor.moveToNext()) {  
        String id = cursor.getString(cursor  
            .getColumnIndex(ContactsContract.Contacts._ID)); // id  
        String name = cursor.getString(cursor  
            .getColumnIndex(ContactsContract.Contacts.DISPLAY_NAME));  
  
        Map<String, String> map = new HashMap<String, String>();  
        map.put("id", id);  
        map.put("name", name);  
        mylist.add(map);  
    }  
    cursor.close(); // 关闭  
    return mylist;  
}  
  
/**  
 * listview中点击按键弹出对话框  
 */  
public void showInfo() {  
    new AlertDialog.Builder(this).setTitle("我的listview")  
        .setMessage("介绍...")  
        .setPositiveButton("确定", new DialogInterface.OnClickListener()  
            @Override  
            public void onClick(DialogInterface dialog, int which)  
            {}  
        }).show();  
}
```

MyCheckBoxAdapter.java类:

```
package iaiai.test;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.CheckBox;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

public class MyCheckBoxAdapter extends BaseAdapter {

    private LayoutInflater mInflater;

    private List<Map<String, String>> listData;

    private Map<Integer, Map<String, String>> selectMap = new HashMap<Integer, Map<String, String>>();

    private class ViewHolder {
        public ImageView img;
        public TextView title;
        public CheckBox checkBox;
    }

    public MyCheckBoxAdapter(Context context, List<Map<String, String>> listData) {
        this.mInflater = LayoutInflater.from(context);
        this.listData = listData;
    }
}
```

```
@Override
public int getCount() {
    return listData.size();
}

@Override
public Object getItem(int position) {
    return listData.get(position);
}

@Override
public long getItemId(int position) {
    return position;
}

@Override
public View getView(final int position, View convertView, ViewGroup parent) {
    ViewHolder holder = null;
    if (convertView == null) {
        holder = new ViewHolder();
        convertView = mInflater.inflate(R.layout.list_items, null);
        final View view = convertView;
        holder.img = (ImageView) convertView.findViewById(R.id.img);
        holder.title = (TextView) convertView.findViewById(R.id.itemTitle);
        holder.checkBox = (CheckBox) convertView.findViewById(R.id.cb);
        holder.checkBox.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                if (selectMap.get(position) != null) {
                    selectMap.remove(position);
                } else {
                    selectMap.put(position, listData.get(position));
                }
                Toast.makeText(view.getContext(),
                    "你选择了：" + selectMap.size() + "个。",
                    Toast.LENGTH_LONG).show();
            }
        });
    }
    return convertView;
}
```

```
        }

    });

    convertView.setTag(holder);

} else {

    holder = (ViewHolder) convertView.getTag();

}

holder.img.setImageResource(R.drawable.icon);

holder.title.setText(listData.get(position).get("name"));

if (selectMap.get(position) != null) {

    holder.checkBox.setChecked(true);

} else {

    holder.checkBox.setChecked(false);

}

return convertView;

}

}
```

运行结果:



1.18 Android 进行单元测试

发表时间: 2011-06-01 关键字: 单元测试, Android, 软件测试, junit, XML

在实际开发中, 开发android软件的过程需要不断地进行测试。使用JUnit测试框架, 是正规Android开发的必用技术, 在JUnit中可以得到组件, 可以模拟发送事件和检测程序处理的正确性。

AndroidManifest.xml中加入下面代码:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.iaiai.activity" android:versionCode="1" android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <!-- 添加测试 -->
        <uses-library android:name="android.test.runner" />
        <!-- <activity android:name="com.iaiai.activity.MyList" android:label="@string/
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity> -->
    </application>
    <!-- activity程序的包地址 -->
    <instrumentation android:name="android.test.InstrumentationTestRunner"
        android:targetPackage="com.iaiai.activity" android:label="Tests for My App" />
    <!-- 读取联系人 -->
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <!-- 写入联系人 -->
    <uses-permission android:name="android.permission.WRITE_CONTACTS" />
    <!-- 在SD卡中创建文件与删除文件权限 -->
    <uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
    <!-- 往SD卡中写入数据权限 -->
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <!-- 电话拨号 -->
    <uses-permission android:name="android.permission.CALL_PHONE" />
</manifest>
```

这个文件要注意的地方是manifest的package属性，刚才就犯了一个错，改包名之后没有改这里，所以总会报错，一定要注意这里。

上面targetPackage指定的包要和应用的package相同。

编写单元测试代码Test.java类：

```
package com.iaiai.test;

import android.test.AndroidTestCase;
import android.util.Log;

public class Test extends AndroidTestCase {

    private final String TAG = "Test";

    public void testSave() throws Exception {
        Log.i(TAG, "测试.....");
    }

}
```

1.19 Android 获得本机ip地址和MAC地址

发表时间: 2011-06-01 关键字: Android, Java, .net, ITeYe, XML

在清单文件中加入以下权限AndroidManifest.xml:

引用

```
<!-- 访问网络 -->
<uses-permission android:name="android.permission.INTERNET" />
<!-- 访问Wi-Fi网络 -->
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

获取Android本机IP地址和MAC , Test.java类:

```
package com.iaiai.test;

import java.net.InetAddress;
import java.net.NetworkInterface;
import java.net.SocketException;
import java.util.Enumeration;

import android.content.Context;
import android.net.wifi.WifiInfo;
import android.net.wifi.WifiManager;
import android.test.AndroidTestCase;
import android.util.Log;

public class Test extends AndroidTestCase {

    private final String TAG = "Test";

    public void testLocalIpAndMac() {
        Log.i(TAG, "IP: " + getLocalIpAddress() + ", MAC: "
                + getLocalMacAddress());
    }
}
```

```
/*
 * 获得Android本机IP地址
 *
 * @return
 */
private String getLocalIpAddress() {
    try {
        for (Enumeration<NetworkInterface> en = NetworkInterface
                .getNetworkInterfaces(); en.hasMoreElements();) {
            NetworkInterface intf = en.nextElement();
            for (Enumeration<InetAddress> enumIpAddr = intf
                    .getInetAddresses(); enumIpAddr.hasMoreElements();
            ) {
                InetAddress inetAddress = enumIpAddr.nextElement();
                if (!inetAddress.isLoopbackAddress()) {
                    return inetAddress.getHostAddress().toString();
                }
            }
        }
    } catch (SocketException ex) {
        Log.e("WifiPreference IpAddress", ex.toString());
    }
    return null;
}

/*
 * 获得Android本机MAC
 *
 * @return
 */
private String getLocalMacAddress() {
    WifiManager wifi = (WifiManager) this.getContext().getSystemService(
        Context.WIFI_SERVICE);
    WifiInfo info = wifi.getConnectionInfo();
    return info.getMacAddress();
}

}
```

运行结果:

Log			
Time	pid	tag	Message
06-01 10:05...	I 690	Test	IP: 10.0.2.15, MAC: null

1.20 Android 事件监听

发表时间: 2011-06-01 关键字: Android, ITeye, UP, OS, UI

事件就是用户与UI界面的交互时所触发的操作。比如点击某一个按钮，会触发一个事件等

1、最长用的按钮事件：OnClickListener

```
mybutton = (Button)findViewById(R.id.mybutton);
mybutton.setOnClickListener(new Button.OnClickListener()  {
    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        ShowMessage("点击了Button按钮");
    }
});
```

2、我们还可以自定义KeyEvent，比如

```
public boolean onKeyDown(int keyCode, KeyEvent event)  {
    /**
     * 这里构建KeyEvent对象，功能为返回键的功能
     * 即我们按任意键都是返回键功能
     */
    KeyEvent key = new KeyEvent(KeyEvent.ACTION_DOWN, KeyEvent.KEYCODE_BACK);
    return super.onKeyDown(keyCode, event);
}
```

3、手机按键的事件处理

Boolean onKeyDown(int keyCode, KeyEvent event) : 相应按键按下

Boolean onKeyUp(int keyCode, KeyEvent event) : 相应按键释放

Boolean onKeyMultiple() : 相应按键重复点击

Boolean onTouchEvent(MotionEvent event) : 相应触屏事件

```
package com.iaiai.activity;

import android.app.Activity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.MotionEvent;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class EventTest01 extends Activity {

    /**
     * 注意，要重写onKeyDown, onKeyUp, onTouchEvent方法
     */
    private Button mybutton;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        mybutton = (Button) findViewById(R.id.mybutton);
        mybutton.setOnClickListener(new Button.OnClickListener()
        {

            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                ShowMessage("点击了Button按钮");
            }
        });
    }

    /**
     *按键按下触发的事件*/
    public boolean onKeyDown(int keyCode, KeyEvent event)
    {
        switch(keyCode)
        {
```

```
case KeyEvent.KEYCODE_DPAD_CENTER:
    ShowMessage("按下:中键");break;
case KeyEvent.KEYCODE_DPAD_UP:
    ShowMessage("按下:上方向键");break;
case KeyEvent.KEYCODE_DPAD_DOWN:
    ShowMessage("按下:下方向键");break;
case KeyEvent.KEYCODE_DPAD_LEFT:
    ShowMessage("按下:左方向键");break;
case KeyEvent.KEYCODE_DPAD_RIGHT:
    ShowMessage("按下:右方向键");break;
}
return super.onKeyDown(keyCode, event);
}

/**按键弹起所触发的事件*/
public boolean onKeyUp(int keyCode, KeyEvent event)
{
    switch(keyCode)
    {
        case KeyEvent.KEYCODE_DPAD_CENTER:
            ShowMessage("弹起:中键");
            break;
        case KeyEvent.KEYCODE_DPAD_DOWN:
            ShowMessage("弹起:下方向键");
            break;
        case KeyEvent.KEYCODE_DPAD_UP:
            ShowMessage("弹起:上方向键");
            break;
        case KeyEvent.KEYCODE_DPAD_LEFT:
            ShowMessage("弹起:左方向键");
            break;
        case KeyEvent.KEYCODE_DPAD_RIGHT:
            ShowMessage("弹起:右方向键");
            break;
    }
    return super.onKeyUp(keyCode, event);
}
```

```
/**按键重复点击事件*/
public boolean onKeyMultiple(int keyCode, int repeatCount, KeyEvent event)
{
    return onKeyMultiple(keyCode, repeatCount, event);
}

/**触屏事件*/
public boolean onTouchEvent(MotionEvent event)
{
    int action = event.getAction();
    if(action == MotionEvent.ACTION_CANCEL||action == MotionEvent.ACTION_DOWN||action == MotionEvent.ACTION_UP)
    {
        return false;
    }
    //得到触点的位置
    String x = String.valueOf(event.getX());
    String y = String.valueOf(event.getY());

    ShowMessage("触点坐标 : (" + x + "," + y + ")");
    return super.onTouchEvent(event);
}

/**显示触发事件的信息*/
public void ShowMessage(String str)
{
    Toast toast = Toast.makeText(this, str, Toast.LENGTH_SHORT);
    toast.show();
}
```

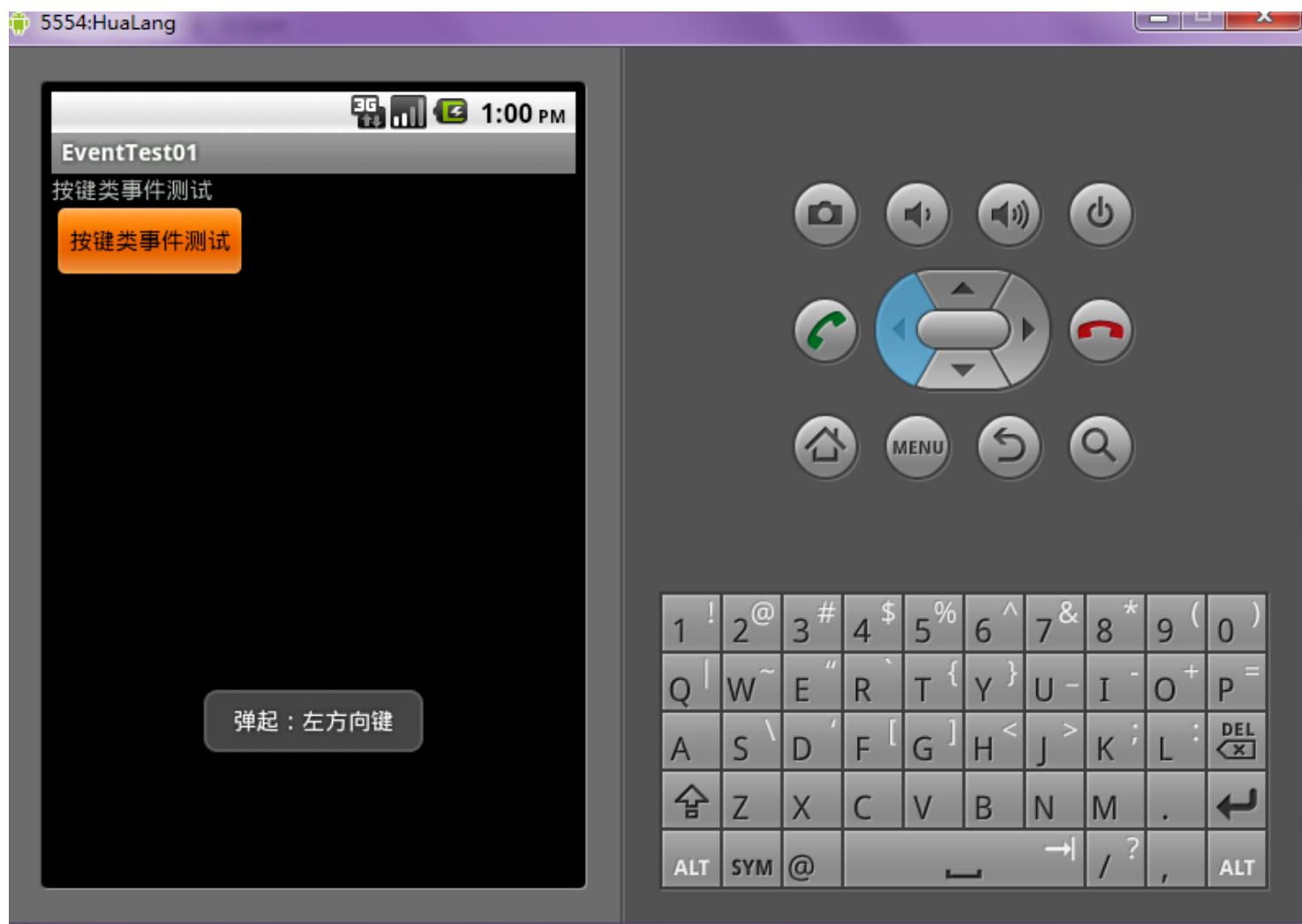
运行结果如下：

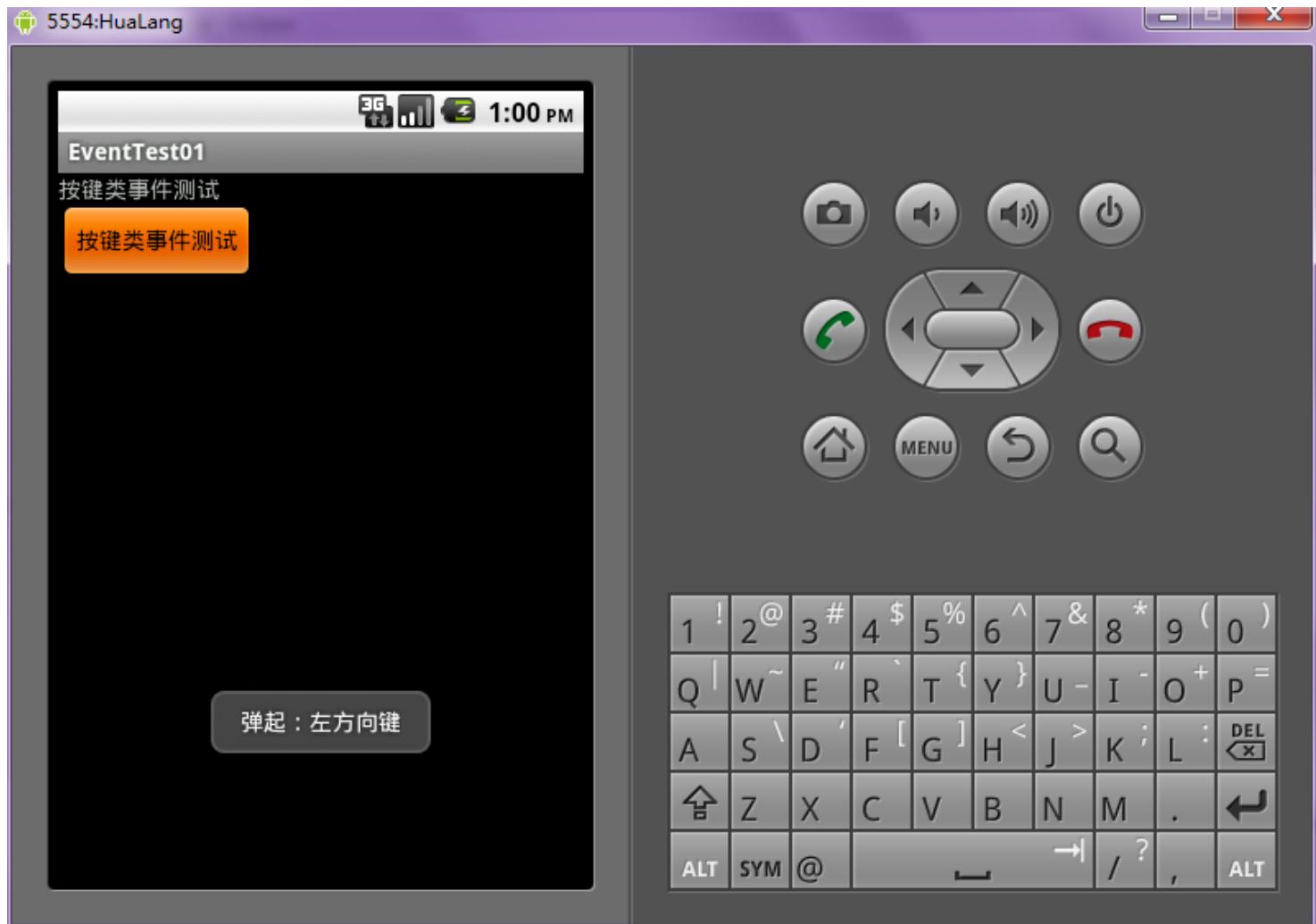












1.21 Android 控件使用参数集锦

发表时间: 2011-06-01 关键字: Android

有新的则更新

属性	值	说明
android:orientation	horizontal/vertical	设置布局水平还是垂直，默认是垂直
android:checked	true/false	标记默认选中，如果是单选则选中最后一个位置
android:layout_gravity	center/right/left/bottom/top	位置(android:gravity属性是对该view内容的限定。比如一个button上面的text.你可以设置该text在view的靠左，靠右等位置。该属性就干了这个。android:layout_gravity是用来设置该view相对与起父view的位置。比如一个button在linearlayout里，你想把该button放在靠左靠右等位置就可以通过该属性设置。)
android:gravity	center/right/left/bottom/top	提示信息，当文本框为空的时候显示只能输入整数/小数
android:hint	@string	设置单行输入，一旦设置为true，则文字不会自动换行
android:numeric	integer/decimal	设置密码框
android:singleLine	true/false	字体颜色
android:password	true/false	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:textColor	#ffffff	字体大小
android:textStyle	bold/italic/bolditalic	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
android:textSize	20dip	设置文字显示位置，EditText没有这个属性，但TextView有
android:capitalize	characters	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:textAlign	center/right/left	是否可编辑
android:autoText	true/false	被选中文字的底色
android:editable	true/false	
android:textColorHighlight	#ffffff	

android:textColorHint	#ffffff	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	normal/sans-serif/monospace	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:background	@null	空间背景，这里没有，指透明，将EditText自定义的背景去掉
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	textAppearanceButton、 textAppearanceInverse、 textAppearanceLarge、 textAppearanceLargeInverse、 textAppearanceMedium、 textAppearanceMediumInverse、 textAppearanceSmall、 textAppearanceSmallInverse	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。
android:autoLink	none、web、email、phone、 map、all	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。
android:cursorVisible	true/false	设定光标为显示/隐藏，默认显示。
android:digits	1234567890.-*/%\n()	设置允许输入哪些字符。
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: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: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:editorExtras		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。

android:ellipsize	start、end、middle、marquee	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	true/false	设置保存文本的内容以及光标的位置,再到相应的Activity中加上如下两个方法,实现方法网站找一下,这里不详细说明。
android:imeOptions	normal, actionUnspecified, actionNone, actionGo, actionSearch, actionSend, actionNext, actionDone, flagNoExtractUi, flagNoAccessoryAction, flagNoEnterAction	设置软键盘的Enter键。可用‘ ’设置多个。
android:imeActionId		设置IME动作ID,在onEditorAction中捕获判断进行逻辑操作。
android:imeActionButton		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。
android:includeFontPadding	true/false	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)
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。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)、textLongMessage长讯

android:linksClickable		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、textFilte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。
android:maxLines		设置链接是否点击连接，即使设置了autoLink
android:minLines		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:lineSpacingExtra		设置文本的最小行数
android:phoneNumber	true/false	设置行间距。
android:privateImeOptions		设置为电话号码的输入方式。
android:scrollHorizontally	true/false	提供额外的输入法选项(字符串格式)。
android:selectAllOnFocus		设置文本超出TextView的宽度的情况下，是否出现横拉条。
android:shadowColor		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。
android:shadowRadius		指定文本阴影的颜色，需要与shadowRadius一起使用。
android:shadowDx		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:shadowDy		设置阴影横向坐标开始位置。
android:textColorLink	#ffffff	设置阴影纵向坐标开始位置。
android:maxHeight		文字链接的颜色。
android:minHeight		设置文本区域的最大高度
android:maxWidth		设置文本区域的最小高度
android:minWidth		设置文本区域的最大宽度
android:width		设置文本区域的最小宽度
android:clickable	true/false	view是否能对click事件作出反应

android:onClick

android:longClickable true/false

android:focusable

android:cacheColorHint #00000000

当view被click之后，view的context的那个方法被呼叫。通常这个context是指view所在的Activity。例

如:android:onClick='sayHello'，则相应的Activity里面有一个方法public void sayHello(View view)方法，当这个view被click之后，sayHello方法就会被调用。
view是否可以对长时间的click事件作出反应。

view是否能响应焦点事件

滚动时，列表里面的view重绘时，用的依旧是系统默认的透明色，颜色值为#FF191919，要改变这种情况，只需要调用listView的setCacheColorHint(0)，颜色值设置为0或者xml文件中listView的属性Android:cacheColorHint="#00000000"即可，滚动时，重绘View的时候就不会有背景颜色。

属性值为true或false

值	说明
android:layout_centerHorizontal	水平居中
android:layout_centerVertical	垂直居中
android:layout_centerInparent	相对于父元素完全居中
android:layout_alignParentBottom	贴紧父元素的下边缘
android:layout_alignParentLeft	贴紧父元素的左边缘
android:layout_alignParentRight	贴紧父元素的右边缘
android:layout_alignParentTop	贴紧父元素的上边缘
android:layout_alignWithParentIfMissing	如果对应的兄弟元素找不到的话就以父元素做参照物

属性值必须为id的引用名"@id/id-name"

值	说明
android:layout_below	在某元素的下方
android:layout_above	在某元素的上方

android:layout_toLeftOf	在某元素的左边
android:layout_toRightOf	在某元素的右边
android:layout_alignTop	本元素的上边缘和某元素的上边缘对齐
android:layout_alignLeft	本元素的左边缘和某元素的左边缘对齐
android:layout_alignBottom	本元素的下边缘和某元素的下边缘对齐
android:layout_alignRight	本元素的右边缘和某元素的右边缘对齐

属性值为具体的像素值，如30dip，40px

值	说明
android:layout_marginBottom	离某元素底边缘的距离
android:layout_marginLeft	离某元素左边缘的距离
android:layout_marginRight	离某元素右边缘的距离
android:layout_marginTop	离某元素上边缘的距离

1.22 Android HttpClient网络通信

发表时间: 2011-06-08 关键字: Android, Apache, Java, OS

封装Get/Post两种提交的方法类 ,

HttpUtil.java类:

```
package com.water.activity;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.HttpStatus;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.params.CoreConnectionPNames;
import org.apache.http.util.EntityUtils;

public class HttpUtil {

    /**
     * http访问数据 , 默认返回的数据为UTF-8编码
     *
     * @param url
     *          url地址
     * @param method
     */
```

```
/*
 *      提交方式
 */
* @return
* @throws IOException
*/
public static String getUrl(String url) throws IOException {
    HttpGet request = new HttpGet(url);
    HttpClient httpClient = new DefaultHttpClient();
    HttpResponse response = httpClient.execute(request);
    if (response.getStatusLine().getStatusCode() == HttpStatus.SC_OK) {
        return EntityUtils.toString(response.getEntity());
    } else {
        return "";
    }
}

/**
 * post提交数据
 *
 * @param url
 *      提交地址
 * @param params
 *      参数
 * @return
 * @throws ClientProtocolException
 * @throws IOException
*/
public static String postUrl(String url, Map<String, String> params)
    throws ClientProtocolException, IOException {
    return postUrl(url, params, "UTF-8");
}

/**
 * post提交数据
 *
 * @param url
 *      提交地址
 * @param params

```

```
*          参数
* @param encoding
*          参数编码
* @return
* @throws ClientProtocolException
* @throws IOException
*/
public static String postUrl(String url, Map<String, String> params,
        String encoding) throws ClientProtocolException, IOException {
    List<NameValuePair> param = new ArrayList<NameValuePair>(); // 参数
    // param.add(new BasicNameValuePair("par", "request-post"));
    // //这个参数我不知道是不是非要加这个

    // 添加参数
    Iterator<Entry<String, String>> iterator = params.entrySet().iterator();
    while (iterator.hasNext()) {
        Entry<String, String> entry = iterator.next();

        param.add(new BasicNameValuePair(entry.getKey(), entry.getValue()));
    }

    HttpPost request = new HttpPost(url);
    HttpEntity entity = new UrlEncodedFormEntity(param, encoding);
    request.setEntity(entity);

    HttpClient client = new DefaultHttpClient();
    client.getParams().setParameter(
            CoreConnectionPNames.CONNECTION_TIMEOUT, 10000); // 设置请求超时
    client.getParams().setParameter(CoreConnectionPNames.SO_TIMEOUT, 10000); // 读取

    HttpResponse response = client.execute(request);
    if (response.getStatusLine().getStatusCode() == HttpStatus.SC_OK) {
        return EntityUtils.toString(response.getEntity());
    } else {
        return null;
    }
}
```

}

MyList.java类:

```
package com.water.activity;

import java.util.HashMap;
import java.util.Map;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.widget.EditText;

public class MyList extends Activity {

    private EditText editView;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        try {
            Map map = new HashMap<String, String>();
            map.put("name", "name");
            map.put("password", "password");

            String content = HttpUtil.postUrl("http://www.163.com", map);
            if (content == null) {
                Log.i("MyList", "返回值有问题!");
            } else {
                Log.i("MyList", content);
            }
        } catch (Exception e) {
```

```
        Log.i("MyList", "运行出错!");  
    }  
}  
  
}
```

1.23 Android 读取SIM卡参数

发表时间: 2011-06-10 关键字: Android, 中国移动, 中国电信, Mobile, 电信

```
package com.water.activity;

import java.util.List;

import android.app.Activity;
import android.os.Bundle;
import android.telephony.CellLocation;
import android.telephony.NeighboringCellInfo;
import android.telephony.TelephonyManager;
import android.util.Log;

public class MyList extends Activity {

    /**
     * android API中的TelephonyManager对象，可以取得SIM卡中的信息
     */
    private TelephonyManager telMgr;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        telMgr = (TelephonyManager) getSystemService(TELEPHONY_SERVICE);

        if (telMgr.getSimState() == telMgr.SIM_STATE_READY) {
            Log.i("MyList", "良好");
        } else if (telMgr.getSimState() == telMgr.SIM_STATE_ABSENT) {
            Log.i("MyList", "无SIM卡");
        } else {
            Log.i("MyList", "SIM卡被锁定或未知的状态");
        }
    }
}
```

```
Log.i("MyList", "电话状态[0 无活动/1 响铃/2 摘机]:" + getCallState());
Log.i("MyList", "电话方位:" + getCellLocation());
Log.i("MyList", "唯一的设备ID:" + getDeviceId());
Log.i("MyList", "设备的软件版本号:" + getDeviceSoftwareVersion());
Log.i("MyList", "手机号:" + getLine1Number());
Log.i("MyList", "附近的电话的信息:" + getNeighboringCellInfo());
Log.i("MyList", "获取ISO标准的国家码，即国际长途区号:" + getNetworkCountryIso());
Log.i("MyList", "MCC+MNC:" + getNetworkOperator());
Log.i("MyList", "(当前已注册的用户)的名字:" + getNetworkOperatorName());
Log.i("MyList", "当前使用的网络类型:" + getNetworkType());
Log.i("MyList", "手机类型:" + getPhoneType());
Log.i("MyList", "SIM卡的国家码:" + getSimCountryIso());
Log.i("MyList", "获取SIM卡提供的移动国家码和移动网络码.5或6位的十进制数字:" + getSi
Log.i("MyList", "服务商名称:" + getSimOperatorName());
Log.i("MyList", "SIM卡的序列号:" + getSimSerialNumber());
Log.i("MyList", "SIM的状态信息:" + getSimState());
Log.i("MyList", "唯一的用户ID:" + getSubscriberId());
Log.i("MyList", "取得和语音邮件相关的标签，即为识别符:" + getVoiceMailAlphaTag());
Log.i("MyList", "获取语音邮件号码:" + getVoiceMailNumber());
Log.i("MyList", "ICC卡是否存在:" + hasIccCard());
Log.i("MyList", "是否漫游:" + isNetworkRoaming());
Log.i("MyList", "获取数据活动状态:" + getDataActivity());
Log.i("MyList", "获取数据连接状态:" + getDataState());

// telMgr.listen(PhoneStateListener listener, int events) ;
// 解释：
// IMSI是国际移动用户识别码的简称(International Mobile Subscriber Identity)
// IMSI共有15位，其结构如下：
// MCC+MNC+MIN
// MCC : Mobile Country Code，移动国家码，共3位，中国为460;
// MNC:Mobile NetworkCode，移动网络码，共2位
// 在中国，移动的代码为00和02，联通的代码为01，电信的代码为03
// 合起来就是（也是Android手机中APN配置文件中的代码）：
// 中国移动：46000 46002
// 中国联通：46001
// 中国电信：46003
// 举例，一个典型的IMSI号码为460030912121001
```

```
// IMEI是International Mobile Equipment Identity (国际移动设备标识)的简称
// IMEI由15位数字组成的“电子串号”，它与每台手机一一对应，而且该码是全世界唯一的
// 其组成为：
// 1. 前6位数(TAC)是“型号核准号码”，一般代表机型
// 2. 接着的2位数(FAC)是“最后装配号”，一般代表产地
// 3. 之后的6位数(SNR)是“串号”，一般代表生产顺序号
// 4. 最后1位数(SP)通常是“0”，为检验码，目前暂备用
}

/**
 * 电话状态：<br/>
 * CALL_STATE_IDLE 无任何状态时<br/>
 * CALL_STATE_OFFHOOK 接起电话时<br/>
 * CALL_STATE_RINGING 电话进来时
 *
 * @return
 */
private int getCallState() {
    return telMgr.getCallState();
}

/**
 * 返回当前移动终端的位置 <br/>
 *
 * @return
 */
private CellLocation getCellLocation() {
    CellLocation location = telMgr.getCellLocation();

    // 请求位置更新，如果更新将产生广播，接收对象为注册LISTEN_CELL_LOCATION的对象，需要的
    // location.requestLocationUpdate();

    return location;
}

/**
 * 唯一的设备ID：<br/>

```

```
* 如果是GSM网络，返回IMEI；如果是CDMA网络，返回MEID<br/>
* 需要权限：android.permission.READ_PHONE_STATE
*
* @return null if device ID is not available.
*/
private String getDeviceId() {
    return telMgr.getDeviceId();
}

/**
* 返回移动终端的软件版本：<br/>
* 例如：GSM手机的IMEI/SV码。<br/>
*
* @return null if the software version is not available.
*/
private String getDeviceSoftwareVersion() {
    return telMgr.getDeviceSoftwareVersion();
}

/**
* 手机号：<br/>
* 对于GSM网络来说即MSISDN
*
* @return null if it is unavailable.
*/
private String getLine1Number() {
    return telMgr.getLine1Number();
}

/**
* 返回当前移动终端附近移动终端的信息:<br/>
* 类型：List<NeighboringCellInfo><br/>
* 需要权限：android.Manifest.permission#ACCESS_COARSE_UPDATES
*
* @return
*/
private List<NeighboringCellInfo> getNeighboringCellInfo() {
```

```
// List<NeighboringCellInfo> infos = telMgr.getNeighboringCellInfo();
// for (NeighboringCellInfo info : infos) {
// // 获取邻居小区号
// int cid = info.getCid();
// //
// // 获取邻居小区LAC , LAC:
// // 位置区域码。为了确定移动台的位置，每个GSM/PLMN的覆盖区都被划分成许多位置区，LAC
// info.getLac();
// info.getNetworkType();
// info.getPsc();
// //
// // 获取邻居小区信号强度
// info.getRssi();
// }

return telMgr.getNeighboringCellInfo();
}

/**
 * 获取ISO标准的国家码，即国际长途区号。<br/>
 * 注意：仅当用户已在网络注册后有效。<br/>
 * 在CDMA网络中结果也许不可靠。<br/>
 *
 * @return
 */
private String getNetworkCountryIso() {
    return telMgr.getNetworkCountryIso();
}

/**
 * MCC+MNC(mobile country code + mobile network code)<br/>
 * 注意：仅当用户已在网络注册时有效。<br/>
 * 在CDMA网络中结果也许不可靠。<br/>
 *
 * @return
 */
private String getNetworkOperator() {
```

```
        return telMgr.getNetworkOperator();
    }

    /**
     * 按照字母次序的current registered operator(当前已注册的用户)的名字<br/>
     * 注意：仅当用户已在网络注册时有效。<br/>
     * 在CDMA网络中结果也许不可靠。
     *
     * @return
     */

    private String getNetworkOperatorName() {
        return telMgr.getNetworkOperatorName();
    }

    /**
     * 当前使用的网络类型：<br/>
     * NETWORK_TYPE_UNKNOWN 网络类型未知 0<br/>
     * NETWORK_TYPE_GPRS GPRS网络 1<br/>
     * NETWORK_TYPE_EDGE EDGE网络 2<br/>
     * NETWORK_TYPE_UMTS UMTS网络 3<br/>
     * NETWORK_TYPE_HSDPA HSDPA网络 8<br/>
     * NETWORK_TYPE_HSUPA HSUPA网络 9<br/>
     * NETWORK_TYPE_HSPA HSPA网络 10<br/>
     * NETWORK_TYPE_CDMA CDMA网络, IS95A 或 IS95B. 4<br/>
     * NETWORK_TYPE_EVDO_0 EVDO网络, revision 0. 5<br/>
     * NETWORK_TYPE_EVDO_A EVDO网络, revision A. 6<br/>
     * NETWORK_TYPE_1xRTT 1xRTT网络 7<br/>
     * 在中国，联通的3G为UMTS或HSDPA，移动和联通的2G为GPRS或EGDE，电信的2G为CDMA，电信的3G为EVDO
     *
     * @return
     */

    private int getNetworkType() {
        return telMgr.getNetworkType();
    }

    /**
     * 返回移动终端的类型：<br/>
```

```
* PHONE_TYPE_CDMA 手机制式为CDMA , 电信<br/>
* PHONE_TYPE_GSM 手机制式为GSM , 移动和联通<br/>
* PHONE_TYPE_NONE 手机制式未知<br/>
*
* @return
*/
private int getPhoneType() {
    return telMgr.getPhoneType();
}

/**
 * 获取ISO国家码 , 相当于提供SIM卡的国家码。
*
* @return Returns the ISO country code equivalent for the SIM provider's
*         country code.
*/
private String getSimCountryIso() {
    return telMgr.getSimCountryIso();
}

/**
 * 获取SIM卡提供的移动国家码和移动网络码.5或6位的十进制数字.<br/>
 * SIM卡的状态必须是 SIM_STATE_READY(使用getSimState()判断).
*
* @return Returns the MCC+MNC (mobile country code + mobile network code)
*         of the provider of the SIM. 5 or 6 decimal digits.
*/
private String getSimOperator() {
    return telMgr.getSimOperator();
}

/**
 * 服务商名称 :<br/>
 * 例如 :中国移动、联通<br/>
 * SIM卡的状态必须是 SIM_STATE_READY(使用getSimState()判断).
*
* @return
*/
```

```
/*
private String getSimOperatorName() {
    return telMgr.getSimOperatorName();
}

/***
 * SIM卡的序列号 : <br/>
 * 需要权限 : READ_PHONE_STATE
 *
 * @return
 */
private String getSimSerialNumber() {
    return telMgr.getSimSerialNumber();
}

/***
 * SIM的状态信息 : <br/>
 * SIM_STATE_UNKNOWN 未知状态 0<br/>
 * SIM_STATE_ABSENT 没插卡 1<br/>
 * SIM_STATE_PIN_REQUIRED 锁定状态 , 需要用户的PIN码解锁 2<br/>
 * SIM_STATE_PUK_REQUIRED 锁定状态 , 需要用户的PUK码解锁 3<br/>
 * SIM_STATE_NETWORK_LOCKED 锁定状态 , 需要网络的PIN码解锁 4<br/>
 * SIM_STATE_READY 就绪状态 5
 *
 * @return
 */
private int getSimState() {
    return telMgr.getSimState();
}

/***
 * 唯一的用户ID : <br/>
 * 例如 : IMSI(国际移动用户识别码) for a GSM phone.<br/>
 * 需要权限 : READ_PHONE_STATE
 *
 * @return
 */
```

```
private String getSubscriberId() {  
    return telMgr.getSubscriberId();  
}  
  
/**  
 * 取得和语音邮件相关的标签，即为识别符<br/>  
 * 需要权限：READ_PHONE_STATE  
 *  
 * @return  
 */  
private String getVoiceMailAlphaTag() {  
    return telMgr.getVoiceMailAlphaTag();  
}  
  
/**  
 * 获取语音邮件号码：<br/>  
 * 需要权限：READ_PHONE_STATE  
 *  
 * @return  
 */  
private String getVoiceMailNumber() {  
    return telMgr.getVoiceMailNumber();  
}  
  
/**  
 * ICC卡是否存在  
 *  
 * @return  
 */  
private boolean hasIccCard() {  
    return telMgr.hasIccCard();  
}  
  
/**  
 * 是否漫游：(在GSM用途下)  
 *  
 * @return  
 */
```

```
/*
private boolean isNetworkRoaming() {
    return telMgr.isNetworkRoaming();
}

/***
 * 获取数据活动状态<br/>
 * DATA_ACTIVITY_IN 数据连接状态：活动，正在接受数据<br/>
 * DATA_ACTIVITY_OUT 数据连接状态：活动，正在发送数据<br/>
 * DATA_ACTIVITY_INOUT 数据连接状态：活动，正在接受和发送数据<br/>
 * DATA_ACTIVITY_NONE 数据连接状态：活动，但无数据发送和接受<br/>
 *
 * @return
 */
private int getDataActivity() {
    return telMgr.getDataActivity();
}

/***
 * 获取数据连接状态<br/>
 * DATA_CONNECTED 数据连接状态：已连接<br/>
 * DATA_CONNECTING 数据连接状态：正在连接<br/>
 * DATA_DISCONNECTED 数据连接状态：断开<br/>
 * DATA_SUSPENDED 数据连接状态：暂停<br/>
 *
 * @return
 */
private int getDataState() {
    return telMgr.getDataState();
}

}
```

1.24 Android之permission权限列表

发表时间: 2011-06-10 关键字: Android, 网络应用, Access, WAP, Windows Phone

属性

android.permission.ACCESS_CHECKIN_PROPERTIES

android.permission.ACCESS_COARSE_LOCATION

android.permission.ACCESS_FINE_LOCATION

android.permission.ACCESS_LOCATION_EXTRA_COMMANDS

android.permission.ACCESS_MOCK_LOCATION

android.permission.ACCESS_NETWORK_STATE

android.permission.ACCESS_SURFACE_FLINGER

android.permission.ACCESS_WIFI_STATE

android.permission.ADD_SYSTEM_SERVICE

android.permission.BATTERY_STATS

android.permission.BLUETOOTH

android.permission.BLUETOOTH_ADMIN

android.permission.BRICK

android.permission.BROADCAST_PACKAGE_REMOVED

android.permission.BROADCAST_STICKY

android.permission.CALL_PHONE

android.permission.CALL_PRIVILEGED

android.permission.CAMERA

android.permission.CHANGE_COMPONENT_ENABLED_STATE

android.permission.CHANGE_CONFIGURATION

android.permission.CHANGE_NETWORK_STATE

android.permission.CHANGE_WIFI_STATE

android.permission.CLEAR_APP_CACHE

android.permission.CLEAR_APP_USER_DATA

android.permission.CONTROL_LOCATION_UPDATES

android.permission.DELETE_CACHE_FILES

说明

允许读写访问 "properties"表在checkin数据库中
值可以修改上传

允许一个程序访问CellID或WiFi热点来获取粗略的
位置

允许一个程序访问精良位置(如GPS)

允许应用程序访问额外的位置提供命令

允许程序创建模拟位置提供用于测试

允许程序访问有关GSM网络信息

允许程序使用SurfaceFlinger底层特性

允许程序访问Wi-Fi网络状态信息

允许程序发布系统级服务

允许程序更新手机电池统计信息

允许程序连接到已配对的蓝牙设备

允许程序发现和配对蓝牙设备

请求能够禁用设备(非常危险)

允许程序广播一个提示消息在一个应用程序包已
经删除后

允许一个程序广播常用intents

允许一个程序初始化一个电话拨号不需通过拨号
界面需要用户确认

允许一个程序拨打任何号码，包含紧 急号码无需
拨号用户界面需要用户确认

请求访问使用照相设备

允许一个程序是否改变一个组件或其他的启用或禁
用

允许一个程序修改当前设置，如本地化

允许程序改变网络连接状态

允许程序改变Wi-Fi连接状态

允许一个程序清楚缓存从所有安装的程序在设备中

允许一个程序清除用户设置

允许启用禁止位置更新提示从无线模块

允许程序删除缓存文件

android.permission.DELETE_PACKAGES	允许一个程序删除包
android.permission.DEVICE_POWER	允许访问底层电源管理
android.permission.DIAGNOSTIC	允许程序RW诊断资源
android.permission.DISABLE_KEYGUARD	允许程序禁用键盘锁
android.permission.DUMP	允许程序返回状态抓取信息从系统服务
android.permission.EXPAND_STATUS_BAR	允许一个程序扩展收缩在状态栏,Android开发网扯
android.permission.FACTORY_TEST	应该是一个类似Windows Mobile中的托盘程序
android.permission.FLASHLIGHT	作为一个工厂测试程序,运行在root用户
android.permission.FORCE_BACK	访问闪光灯,Android开发网提示HTC Dream不包
android.permission.FOTA_UPDATE	光灯
android.permission.GET_ACCOUNTS	允许程序强行一个后退操作是否在顶层activities
android.permission.GET_PACKAGE_SIZE	暂时不了解这是做什么使用的,Android开发网分
android.permission.GET_TASKS	可能是一个预留权限.
android.permission.HARDWARE_TEST	访问一个帐户列表在Accounts Service中
android.permission.INJECT_EVENTS	允许一个程序获取任何package占用空间容量
android.permission.INSTALL_PACKAGES	允许一个程序获取信息有关当前或最近运行的任
android.permission.INTERNAL_SYSTEM_WINDOW	一个缩略的任务状态,是否活动等等
android.permission.INTERNET	允许访问硬件
android.permission.MANAGE_APP_TOKENS	允许一个程序截获用户事件如按键、触摸、轨迹
android.permission.MASTER_CLEAR	等到一个时间流
android.permission.MODIFY_AUDIO_SETTINGS	允许一个程序安装packages
android.permission.MODIFY_PHONE_STATE	允许打开窗口使用系统用户界面
android.permission.MOUNT_UNMOUNT_FILESYSTEMS	允许程序打开网络套接字
android.permission.PERSISTENT_ACTIVITY	允许程序管理(创建、催后、z-order默认向z轴推
android.permission.PROCESS_OUTGOING_CALLS	程序引用在窗口管理器中
android.permission.READ_CALENDAR	目前还没有明确的解释,可能是清除一切数据,关
android.permission.READ_CONTACTS	硬格机
android.permission.READ_FRAME_BUFFER	允许程序修改全局音频设置
	允许修改话机状态,如电源,人机接口等
	允许挂载和反挂载文件系统可移动存储
	允许一个程序设置他的activities显示
	允许程序监视、修改有关播出电话
	允许程序读取用户日历数据
	允许程序读取用户联系人数据
	允许程序屏幕波或和更多常规的访问帧缓冲数据

android.permission.READ_INPUT_STATE	允许程序返回当前按键状态
android.permission.READ_LOGS	允许程序读取底层系统日志文件
android.permission.READ_OWNER_DATA	允许程序读取所有者数据
android.permission.READ_SMS	允许程序读取短信息
android.permission.READ_SYNC_SETTINGS	允许程序读取同步设置
android.permission.READ_SYNC_STATS	允许程序读取同步状态
android.permission.REBOOT	请求能够重新启动设备
android.permission.RECEIVE_BOOT_COMPLETED	允许一个程序接收到 ACTION_BOOT_COMPLETED 广播在系统完成启动
android.permission.RECEIVE_MMS	允许一个程序监控将收到MMS彩信,记录或处理
android.permission.RECEIVE_SMS	允许程序监控一个将收到短信息 , 记录或处理
android.permission.RECEIVE_WAP_PUSH	允许程序监控将收到WAP PUSH信息
android.permission.RECORD_AUDIO	允许程序录制音频
android.permission.REORDER_TASKS	允许程序改变Z轴排列任务
android.permission.RESTART_PACKAGES	允许程序重新启动其他程序
android.permission.SEND_SMS	允许程序发送SMS短信
android.permission.SET_ACTIVITY_WATCHER	允许程序监控或控制activities已经启动全局系统中
android.permission.SET_ALWAYS_FINISH	允许程序控制是否活动间接完成在处于后台时
android.permission.SET_ANIMATION_SCALE	修改全局信息比例
android.permission.SET_DEBUG_APP	配置一个程序用于调试
android.permission.SET_ORIENTATION	允许底层访问设置屏幕方向和实际旋转
android.permission.SET_PREFERRED_APPLICATIONS	允许一个程序修改列表参数 PackageManager.addPackageToPreferred() 和 packageManager.removePackageFromPreferred() 方法
android.permission.SET_PROCESS_FOREGROUND	允许程序当前运行程序强行到前台
android.permission.SET_PROCESS_LIMIT	允许设置最大的运行进程数量
android.permission.SET_TIME_ZONE	允许程序设置时间区域
android.permission.SET_WALLPAPER	允许程序设置壁纸
android.permission.SET_WALLPAPER_HINTS	允许程序设置壁纸hints
android.permission.SIGNAL_PERSISTENT_PROCESSES	允许程序请求发送信号到所有显示的进程中
android.permission.STATUS_BAR	允许程序打开、关闭或禁用状态栏及图标
android.permission.SUBSCRIBED_FEEDS_READ	允许一个程序访问订阅RSS Feed内容提供
android.permission.SUBSCRIBED_FEEDS_WRITE	系统暂时保留改设置,未来版本可能会加入该功能

android.permission.SYSTEM_ALERT_WINDOW	允许一个程序打开窗口使用 TYPE_SYSTEM_ALERT 显示在其他所有程序的顶层
android.permission.VIBRATE	允许访问振动设备
android.permission.WAKE_LOCK	允许使用PowerManager的 WakeLocks保持进程 休眠时从屏幕消失
android.permission.WRITE_APN_SETTINGS	允许程序写入API设置
android.permission.WRITE_CALENDAR	允许一个程序写入但不读取用户日历数据
android.permission.WRITE_CONTACTS	允许程序写入但不读取用户联系人数据
android.permission.WRITE_GSERVICES	允许程序修改Google服务地图
android.permission.WRITE_OWNER_DATA	允许一个程序写入但不读取所有者数据
android.permission.WRITE_SETTINGS	允许程序读取或写入系统设置
android.permission.WRITE_SMS	允许程序写短信
android.permission.WRITE_SYNC_SETTINGS	允许程序写入同步设置

1.25 Android开发中如何实现开机自启动

发表时间: 2011-06-11 关键字: Android, 网络应用, XML

如果您在开发一个需要实时更新数据的应用程序，当有新的数据的时候提醒用户查看新的数据，那么您需要在后台开起一个Service，然后实时的去网络上获取数据，但是如果用户关机重启，您的Service可能就消失了！那么怎么样保证开机后你的Service还活跃的在用户的手机里偷偷的从网络上获取数据呢？

很简单，我们只要实现开机自启动即可，android实现开机自启动可能是移动操作系统中最简单的了，我们只需要监听一个开机启动的Broadcast（广播）即可。首先写一个Receiver（即广播监听器），继承BroadcastReceiver，如下所示：

```
public class BootReceiver extends BroadcastReceiver {  
    private PendingIntent mAlarmSender;  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        // 在这里干你想干的事（启动一个Service，Activity等），本例是启动一个定时调度程序，每:  
        mAlarmSender = PendingIntent.getService(context, 0, new Intent(context,  
                RefreshDataService.class), 0);  
        long firstTime = SystemClock.elapsedRealtime();  
        AlarmManager am = (AlarmManager) context  
                .getSystemService(Activity.ALARM_SERVICE);  
        am.cancel(mAlarmSender);  
        am.setRepeating(AlarmManager.ELAPSED_REALTIME_WAKEUP, firstTime,  
                30 * 60 * 1000, mAlarmSender);  
    }  
}
```

接下来，我们只需要在应用程序配置文件AndroidManifest.xml中注册这个Receiver来监听系统启动事件即可，如下所示：

```
<receiver android:name=".service.BootReceiver">  
    <intent-filter>  
        <!-- 系统启动完成后会调用-->  
        <action android:name="android.intent.action.BOOT_COMPLETED">  
        </action>  
    </intent-filter>  
</receiver>
```

1.26 Android Activity之间切换的各种效果

发表时间: 2011-06-12 关键字: Android, Java, ITeye, XML, UP

main.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:id="@+id/contactItems"
    android:paddingBottom="5dip" android:paddingLeft="12dip"
    android:paddingTop="5dip" android:paddingRight="12dip">
    <Spinner android:layout_height="wrap_content"
        android:layout_width="fill_parent" android:id="@+id/sprAnim"></Spinner>
    <Button android:layout_height="wrap_content"
        android:layout_width="fill_parent" android:id="@+id/btn_next"
        android:text="@string/next"></Button>
</LinearLayout>
```

next.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:id="@+id/contactItems"
    android:paddingBottom="5dip" android:paddingLeft="12dip"
    android:paddingTop="5dip" android:paddingRight="12dip">
    <TextView android:layout_height="wrap_content"
        android:layout_width="fill_parent" android:id="@+id/text2"
        android:textSize="20dip" android:paddingTop="12px" android:text="@string/text2"
        android:layout_marginLeft="70px">
    </TextView>
    <EditText android:layout_height="wrap_content"
        android:layout_width="fill_parent"></EditText>
</LinearLayout>
```

strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">丸子联系人</string>
    <string name="contact_title">丸子联系人</string>

    <string name="text1">IAIAI 测试</string>
    <string name="text2">测试跳转效果</string>
    <string name="next">下一页</string>
</resources>
```

arrays.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="anim_type">
        <item>淡入淡出效果</item>
        <item>放大淡出效果</item>
        <item>转动淡出效果1</item>
        <item>转动淡出效果2</item>
        <item>左上角展开淡出效果</item>
        <item>压缩变小淡出效果</item>
        <item>右往左推出效果</item>
        <item>下往上推出效果</item>
        <item>左右交叉效果</item>
        <item>放大淡出效果</item>
        <item>缩小效果</item>
        <item>上下交错效果</item>
    </string-array>
</resources>
```

anim文件夹里的文件就不一一列出来了，你可以从附件里下载整个工程查看里面效果文件。

下面是两个类文件，

IaiaiActivity.java类：

```
package com.iaiai.activity;

import java.util.ArrayList;
import java.util.List;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;

/**
 *
 * <p>
 * Title: IaiaiActivity.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-6-12
 * </p>
 *
 * @author 丸子
 * @version 0.0.1
 */
public class IaiaiActivity extends Activity {
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    final Spinner sprAnim = (Spinner) findViewById(R.id.sprAnim);
    // 获取相应对象
    String[] ls = getResources().getStringArray(R.array.anim_type);
    List<String> list = new ArrayList<String>();
    // 获取XML中定义的数组
    for (int i = 0; i < ls.length; i++) {
        list.add(ls[i]);
    }
    ArrayAdapter<String> animType = new ArrayAdapter<String>(this,
        android.R.layout.simple_spinner_item, list);
    animType.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    sprAnim.setAdapter(animType);
    sprAnim.setSelection(0);

    Button btn = (Button) findViewById(R.id.btn_next);
    btn.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent it = new Intent(IaiaiActivity.this, NextActivity.class);
            startActivityForResult(it, 0);

            switch (sprAnim.getSelectedItemPosition()) {
                case 0:
                    overridePendingTransition(R.anim.fade, R.anim.hold);
                    break;
                case 1:
                    overridePendingTransition(R.anim.my_scale_action,
                        R.anim.my_alpha_action);
                    break;
                case 2:
                    overridePendingTransition(R.anim.scale_rotate,
                        R.anim.my_alpha_action);
            }
        }
    });
}
```

```
        break;

    case 3:
        overridePendingTransition(R.anim.scale_translate_rotate,
                                R.anim.my_alpha_action);
        break;

    case 4:
        overridePendingTransition(R.anim.scale_translate,
                                R.anim.my_alpha_action);
        break;

    case 5:
        overridePendingTransition(R.anim.hyperspace_in,
                                R.anim.hyperspace_out);
        break;

    case 6:
        overridePendingTransition(R.anim.push_left_in,
                                R.anim.push_left_out);
        break;

    case 7:
        overridePendingTransition(R.anim.push_up_in,
                                R.anim.push_up_out);
        break;

    case 8:
        overridePendingTransition(R.anim.slide_left,
                                R.anim.slide_right);
        break;

    case 9:
        overridePendingTransition(R.anim.wave_scale,
                                R.anim.my_alpha_action);
        break;

    case 10:
        overridePendingTransition(R.anim.zoom_enter,
                                R.anim.zoom_exit);
        break;

    case 11:
        overridePendingTransition(R.anim.slide_up_in,
                                R.anim.slide_down_out);
        break;
```

```
        }
    }
});  
}  
}
```

NextActivity.java类:

```
package com.iaiai.activity;  
  
import android.app.Activity;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.KeyEvent;  
  
/**  
 *  
 * <p>  
 * Title: NextActivity.java  
 * </p>  
 * <p>  
 * E-Mail: 176291935@qq.com  
 * </p>  
 * <p>  
 * QQ: 176291935  
 * </p>  
 * <p>  
 * <p>  
 * Http: iaiai.iteye.com  
 * </p>  
 * <p>  
 * Create time: 2011-6-12  
 * </p>  
 *  
 * @author 丸子  
 * @version 0.0.1  
*/
```

```
public class NextActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.next);  
    }  
  
    public boolean onKeyDown(int keyCode, KeyEvent event) {  
        if (keyCode == KeyEvent.KEYCODE_BACK && event.getRepeatCount() == 0) {  
            Intent intent = new Intent(NextActivity.this, IaiaiActivity.class);  
            setResult(RESULT_OK, intent);  
            finish();  
            overridePendingTransition(R.anim.slide_up_in, R.anim.slide_down_out);  
            return false;  
        }  
        return false;  
    }  
}
```

运行效果图:





附件下载:

- testAndroid.rar (101.9 KB)
- dl.iteye.com/topics/download/3fdd7b15-196e-32f5-9390-e12456c2245e

1.27 Android 设置横屏/竖屏

发表时间: 2011-06-12 关键字: XML, Android, OS, ITeye

模拟当点击按钮时，使手机朝向发生改变。

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.iaiai.activity" android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name="IaiaiActivity" android:label="@string/contact_title"
            android:configChanges="orientation">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="NextActivity" android:label="@string/contact_title"></a
    </application>
    <!-- 改变手机配置权限 -->
    <uses-permission android:name="android.permission.CHANGE_CONFIGURATION" />
</manifest>
```

*设置手机的朝向，不然无法获取手机的朝向，必须设置这个参数: android:configChanges="orientation"

IaiaiActivity.java类:

```
package com.iaiai.activity;

import android.app.Activity;
import android.content.pm.ActivityInfo;
import android.content.res.Configuration;
```

```
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class IaiaiActivity extends Activity {

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

        onScreenChanged(getResources().getConfiguration().orientation);

        setContentView(R.layout.main);

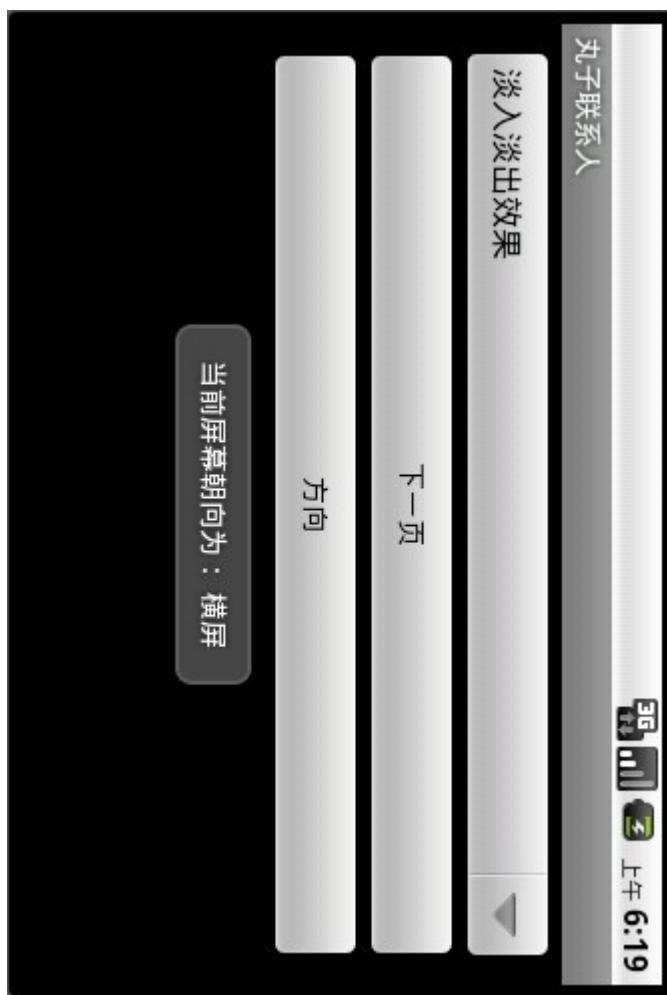
        Button btnfx = (Button) findViewById(R.id.btn_fx);
        btnfx.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                // 判断是否可以获得requestedOrientation属性
                if (getRequestedOrientation() == -1) {
                    Toast.makeText(IaiaiActivity.this, "系统的朝向无法获取",
                        Toast.LENGTH_LONG).show();
                } else {
                    if (getRequestedOrientation() == ActivityInfo.SCREEN_ORIENTATION_PORTRAIT)
                        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
                    } else if (getRequestedOrientation() == ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE)
                        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
                }
            }
        });
    }

    /**
     * 手机横屏
    }
```

```
* @param direction
*/
public void onScreenChanged(int direction) {
    switch (direction) {
        case ActivityInfo.SCREEN_ORIENTATION_USER:
            Log.i("*****", "当前屏幕朝向为： 横屏");
            break;
    }
}

/**
 * 手机竖屏
 *
 * @param newConfig
 */
@Override
public void onConfigurationChanged(Configuration newConfig) {
    switch (newConfig.orientation) {
        case ActivityInfo.SCREEN_ORIENTATION_PORTRAIT:
            Log.i("*****", "当前屏幕朝向为： 竖屏");
            break;
    }
    super.onConfigurationChanged(newConfig);
}
}
```

运行结果:



1.28 Android之快捷方式——通过应用程序创建快捷方式

发表时间: 2011-06-12 关键字: Android, Java, OS, XML

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.iaiai.activity" android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name="IaiaiActivity" android:label="@string/contact_title">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <!-- 添加快捷键权限 -->
    <uses-permission android:name="com.android.launcher.permission.INSTALL_SHORTCUT" />
</manifest>
```

main.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:id="@+id/contactItems"
    android:paddingBottom="5dip" android:paddingLeft="12dip"
    android:paddingTop="5dip" android:paddingRight="12dip">
    <Button android:layout_height="wrap_content"
        android:layout_width="fill_parent" android:id="@+id/btn_create"
        android:text="@string/create"></Button>
</LinearLayout>
```

IaiaiActivity.java类:

```
package com.iaiai.activity;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Parcelable;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class IaiaiActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Button btn = (Button) findViewById(R.id.btn_create);
        btn.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent addIntent = new Intent(
                        "com.android.launcher.action.INSTALL_SHORTCUT");
                Parcelable icon = Intent.ShortcutIconResource.fromContext(
                        IaiaiActivity.this, R.drawable.icon); // 获取快
                Intent myIntent = new Intent(IaiaiActivity.this,
                        IaiaiActivity.class);
                addIntent.putExtra(Intent.EXTRA_SHORTCUT_NAME, "快捷方式");// 快
                addIntent.putExtra(Intent.EXTRA_SHORTCUT_ICON_RESOURCE, icon);
                addIntent.putExtra(Intent.EXTRA_SHORTCUT_INTENT, myIntent); // 快
                sendBroadcast(addIntent); // 发送广播
            }
        });
    }
}
```

1.29 Android之快捷方式二——向Launcher添加快捷方式

发表时间: 2011-06-12 关键字: Android, ITeye, XML

当我们在应用程序Launcher的桌面空白处长按触摸时，会出现一个对话框，提示选择要添加的桌面组件，如下图所示



选择快捷方式后，会弹出一个对话框，显示出了可添加快捷方式的Activity所属的应用程序的图标和名称的列表。当我们想把添加快捷方式的Activity添加到这一列表时，只需要在这个Activity注册时添加一个Action为 android.intent.action.CREATE_SHORTCUT的IntentFilter就可以了。

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.iaiai.activity" android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />
```

```
<application android:icon="@drawable/icon" android:label="@string/app_name">
    <activity android:name="IaiaiActivity" android:label="@string/contact_title">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
        <intent-filter>
            <action
                android:name="android.intent.action.CREATE_SHORTCUT" />
        </intent-filter>
    </activity>
</application>
<!-- 添加快捷键权限 -->
<uses-permission android:name="com.android.launcher.permission.INSTALL_SHORTCUT" />
</manifest>
```



1.30 Android中实现全屏显示的方法

发表时间: 2011-06-12 关键字: Android, XML

在实际的应用程序开发中，我们有时需要把 Activity 设置成全屏显示，一般情况下，可以通过两种方式来设置全屏显示效果。其一，通过在代码中可以设置，其二，通过manifest配置文件来设置全屏。

其一：在代码中设置(如下)

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
  
    //设置无标题  
    requestWindowFeature(Window.FEATURE_NO_TITLE);  
    //设置全屏  
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,  
    WindowManager.LayoutParams.FLAG_FULLSCREEN);  
  
    setContentView(R.layout.main);  
}
```

但要注意的是：在代码中设置的话，设置无标题和设置全屏的两段代码要放置在 setContentView(R.layout.main); 这段代码的前面。要不然会报错。

其二：在AndroidManifest.xml配置文件中设置

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.iaiai.activity" android:versionCode="1"  
    android:versionName="1.0">  
    <uses-sdk android:minSdkVersion="8" />  
    <application android:icon="@drawable/icon" android:label="@string/app_name">  
        <activity android:name="IaiaiActivity"  
            android:theme="@android:style/Theme.NoTitleBar.Fullscreen"  
            android:label="@string/app_name">  
            <intent-filter>
```

```
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
</application>
</manifest>
```

在相应的Activity中节点中添加属性：`android:theme="@android:style/Theme.NoTitleBar.Fullscreen"` 即可以设置某个Activity全屏显示。若设置成 `android:theme="@android:style/Theme.NoTitleBar"` 即是只是设置成无标题状态。

1.31 Android SQLite数据库实例

发表时间: 2011-06-12 关键字: Android, SQLite, ITeye, OS, SQL

DatabaseOpenHelper.java:

```
package com.iaiai.activity;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DatabaseOpenHelper extends SQLiteOpenHelper {

    private static final String DBNAME = "iaiai"; // 数据库名子

    private static final int VERSION = 1; // 版本

    public DatabaseOpenHelper(Context context) {
        // 第三个参数CursorFactory指定在执行查询时获得一个游标实例的工厂类,设置为null,代表使
        super(context, DBNAME, null, VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // 创建表
        db.execSQL("CREATE TABLE ADMIN (ID INTEGER PRIMARY KEY AUTOINCREMENT, NAME VARCHAR");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // 升级更新
        db.execSQL("DROP TABLE IF EXISTS ADMIN");
        onCreate(db);
    }

}
```

IaiaiActivity.java类:

```
package com.iaiai.activity;

import android.app.Activity;
import android.database.Cursor;
import android.os.Bundle;
import android.util.Log;

public class IaiaiActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        DatabaseOpenHelper databaseOpenHelper = new DatabaseOpenHelper(this);
        databaseOpenHelper.getWritableDatabase().execSQL(
            "insert into ADMIN(name, phone) values (?, ?)",
            new Object[] { "用户名1", "111111" });

        Cursor cursor = databaseOpenHelper.getReadableDatabase().query("ADMIN",
            new String[] { "count(*)" }, null, null, null, null, null);
        if (cursor.moveToFirst()) {
            Log.i("*****SQL", "总记录数:" + cursor.getLong(0));
        }

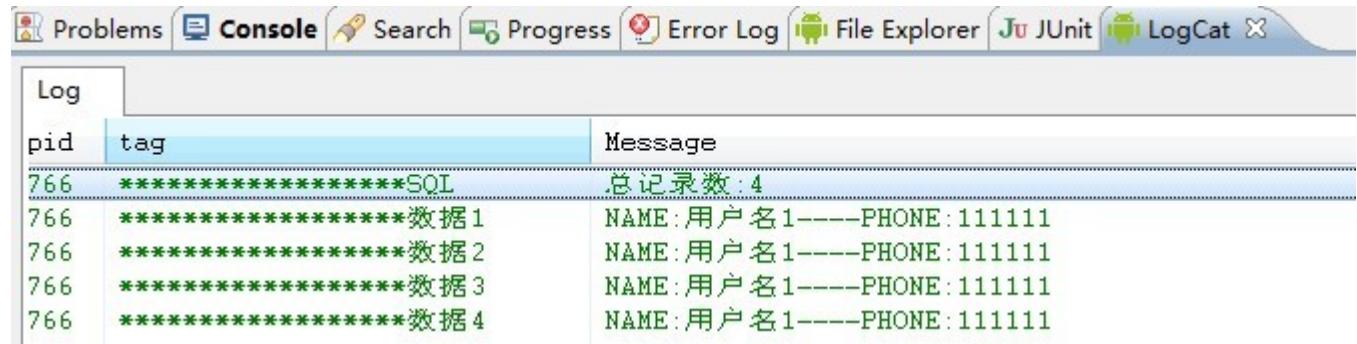
        // cursor = databaseOpenHelper.getReadableDatabase().rawQuery("select ID,NAME,PHONE
        // new String[]{String.valueOf(1)});  

        cursor = databaseOpenHelper.getReadableDatabase().query("ADMIN",null, null, null);
        while(cursor.moveToNext()){
            int adminId = cursor.getInt(0);
            String name = cursor.getString(1);
            String phone = cursor.getString(2);
            Log.i("*****数据"+adminId, "NAME:" + name+"----PHONE:"+phone);
        }
    }
}
```

```
    }  
}
```

只写了最简单的创建数据库，和添加查询数据，其它方法都很简单，这里不再写了。

运行结果：



The screenshot shows the Android Studio LogCat tab. The log entries are as follows:

pid	tag	Message
766	*****SQL	总记录数: 4
766	*****数据1	NAME:用户名1----PHONE:111111
766	*****数据2	NAME:用户名1----PHONE:111111
766	*****数据3	NAME:用户名1----PHONE:111111
766	*****数据4	NAME:用户名1----PHONE:111111

1.32 Android listView 一种常用布局

发表时间: 2011-06-14 关键字: Android, ITeye, Java, XML

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="fill_parent"
    android:layout_height="fill_parent" android:gravity="center_vertical">
    <CheckBox android:id="@+id/checkbox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView android:id="@+id/textView" android:layout_alignParentRight="true"
        android:layout_width="wrap_content" android:layout_height="60dp"
        android:layout_marginLeft="6dp" android:layout_marginRight="6dp"
        android:gravity="center_vertical|right" android:layout_weight="2"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#2B78E4" />
    <TextView android:id="@+id/txt1"
        android:layout_width="fill_parent" android:layout_height="wrap_content"
        android:layout_marginLeft="6dp" android:layout_marginTop="6dp"
        android:layout_toRightOf="@+id/checkbox" android:layout_toLeftOf="@+id/textView"
        android:gravity="center_vertical" android:singleLine="true"
        android:textAppearance="?android:attr/textAppearanceMedium" />
    <TextView android:id="@+id/txt2" android:layout_below="@+id/txt1"
        android:layout_width="fill_parent" android:layout_height="wrap_content"
        android:layout_marginLeft="6dp" android:layout_marginTop="3dp"
        android:layout_toRightOf="@+id/checkbox" android:layout_toLeftOf="@+id/textView"
        android:gravity="center_vertical" android:singleLine="true"
        android:textAppearance="?android:attr/textAppearanceSmall" />
</RelativeLayout>
```

<input checked="" type="checkbox"/>	Lorem ipsum dolor sit amet...	6.2
	Sed ut perspiciatis unde omnis iste n...	
<input checked="" type="checkbox"/>	Lorem ipsum dolor si	6.06 €
	Sed ut perspiciatis unde	
<input checked="" type="checkbox"/>	Lorem ipsum dolor sit am...	12.2
	Sed ut perspiciatis unde omnis iste ...	
<input checked="" type="checkbox"/>	Sed ut perspiciatis unde ...	8.06 €
	Lorem ipsum dolor sit amet, con...	
<input checked="" type="checkbox"/>	Sed ut perspiciatis unde	18.2
	Lorem ipsum dolor si	
<input checked="" type="checkbox"/>	Sed ut perspiciatis und...	10.06 €
	Lorem ipsum dolor sit amet, co...	
<input checked="" type="checkbox"/>	Lorem ipsum dolor sit am...	24.2
	Sed ut perspiciatis unde omnis iste ...	
<input checked="" type="checkbox"/>	Lorem ipsum dolor si	12.06 €
	Sed ut perspiciatis unde	

1.33 Android md5加密

发表时间: 2011-06-14 关键字: Android, Java, Security

```
public String md5(String s) {  
    try {  
        // Create MD5 Hash  
        MessageDigest digest = java.security.MessageDigest.getInstance("MD5");  
        digest.update(s.getBytes());  
        byte messageDigest[] = digest.digest();  
  
        // Create Hex String  
        StringBuffer hexString = new StringBuffer();  
        for (int i=0; i<messageDigest.length; i++)  
            hexString.append(Integer.toHexString(0xFF & messageDigest[i]));  
        return hexString.toString();  
  
    } catch (NoSuchAlgorithmException e) {  
        e.printStackTrace();  
    }  
    return "";  
}  
  
//或者部分加密  
public static String MD5_Hash(String s) { MessageDigest m = null;  
  
    try {  
        m = MessageDigest.getInstance("MD5");  
    } catch (NoSuchAlgorithmException e) {  
        e.printStackTrace();  
    }  
  
    m.update(s.getBytes(),0,s.length());  
    String hash = new BigInteger(1, m.digest()).toString(16);  
    return hash;  
}
```

1.34 Android 画图之Matrix(一)

发表时间: 2011-06-19 关键字: Android, ITeye, OS

Matrix, 中文里叫矩阵, 高等数学里有介绍, 在图像处理方面, 主要是用于平面的缩放、平移、旋转等操作。

首先介绍一下矩阵运算。加法和减法就不用说了, 太简单了, 对应位相加就好。图像处理, 主要用到的是乘法。下面是一个乘法的公式:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \times \begin{pmatrix} x & y \\ m & n \end{pmatrix} = \begin{pmatrix} a*x+b*m & a*y+b*n \\ c*x+d*m & c*y+d*n \end{pmatrix}$$

在Android里面, Matrix由9个float值构成, 是一个3*3的矩阵。如下图。

```
cosX, -sinX, translateX
sinX,  cosX, translateY
0,     0,    scale
```

没专业工具, 画的挺难看。解释一下, 上面的sinX和cosX, 表示旋转角度的cos值和sin值, 注意, 旋转角度是按顺时针方向计算的。translateX和translateY表示x和y的平移量。scale是缩放的比例, 1是不变, 2是表示缩放1/2, 这样子。

IaiaiActivity.java类:

```
package com.iaiai.activity;

import android.app.Activity;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Matrix;
import android.graphics.drawable.BitmapDrawable;
import android.os.Bundle;
import android.view.View;

public class IaiaiActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);

this.setContentView(new MyView(this));

}

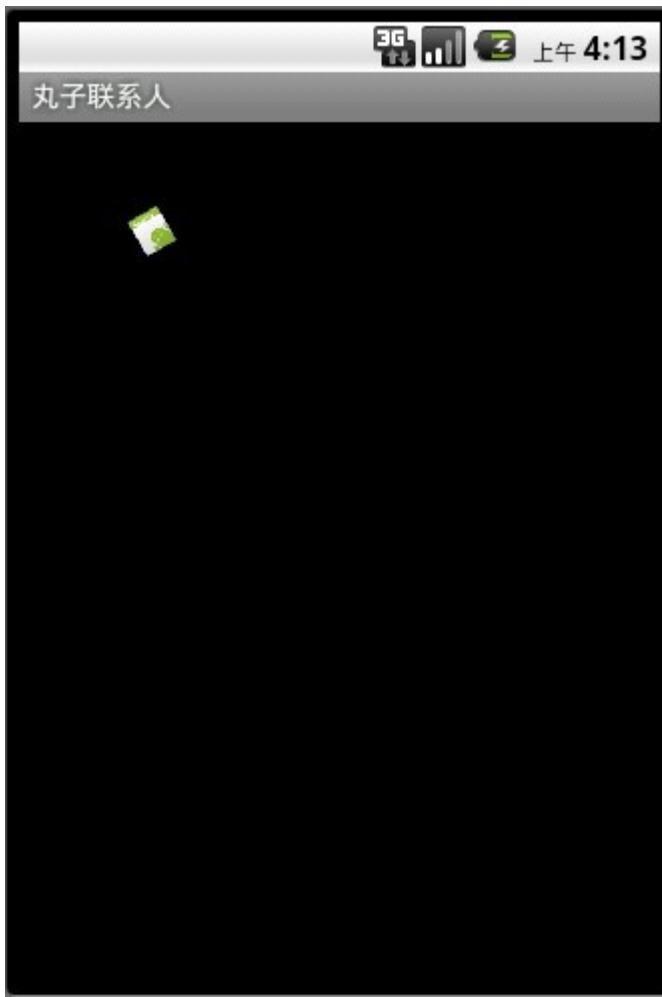
public class MyView extends View {
    private Bitmap bmp;
    private Matrix matrix = new Matrix();

    public MyView(Context context) {
        super(context);
        initialize();
    }

    private void initialize() {
        bmp = ((BitmapDrawable) getResources().getDrawable(R.drawable.icon))
            .getBitmap();
        float cosValue = (float) Math.cos(-Math.PI / 6);
        float sinValue = (float) Math.sin(-Math.PI / 6);
        matrix.setValues(new float[] { cosValue, -sinValue, 100, sinValue,
            cosValue, 100, 0, 0, 2 });
    }

    @Override
    protected void onDraw(Canvas canvas) {
        // super.onDraw(canvas); //当然，如果界面上还有其他元素需要绘制，只需要将这
        canvas.drawBitmap(bmp, matrix, null);
    }
}
```

运行结果：



以左上角为顶点，缩放一半，逆时针旋转30度，然后沿x轴和y轴分别平移50个像素，代码里面写的是100，为什么是平移50呢，因为缩放了一半。

大家可以自己设置一下Matrix的值，或者尝试一下两个Matrix相乘，得到的值设置进去，这样才能对Matrix更加熟练。

这里讲的直接赋值的方式也许有点不好理解，不过还好，android提供了对矩阵的更方便的方法，下一篇介绍。

1.35 Android 画图之Matrix(二)

发表时间: 2011-06-19 关键字: Android, ITeye, OS

上一篇Android画图之Matrix(一)讲了一下Matrix的原理和运算方法，涉及到高等数学，有点难以理解。还好Android里面提供了对Matrix操作的一系列方便的接口。

Matrix的操作，总共分为translate（平移），rotate（旋转），scale（缩放）和skew（倾斜）四种，每一种变换在Android的API里都提供了set,post和pre三种操作方式，除了translate，其他三种操作都可以指定中心点。

set是直接设置Matrix的值，每次set一次，整个Matrix的数组都会变掉。

post是后乘，当前的矩阵乘以参数给出的矩阵。可以连续多次使用post，来完成所需的整个变换。例如，要将一个图片旋转30度，然后平移到(100,100)的地方，那么可以这样做：

```
Matrix m = new Matrix();
m.postRotate(30);
m.postTranslate(100, 100);
```

这样就达到了想要的效果。

pre是前乘，参数给出的矩阵乘以当前的矩阵。所以操作是在当前矩阵的最前面发生的。例如上面的例子，如果用pre的话，就要这样：

```
Matrix m = new Matrix();
m.setTranslate(100, 100);
m.preRotate(30);
```

旋转、缩放和倾斜都可以围绕一个中心点来进行，如果不指定，默认情况下，是围绕(0,0)点来进行。

下面给出一个例子。

```
package com.iaiai.activity;

import android.app.Activity;
import android.content.Context;
import android.graphics.Bitmap;
```

```
import android.graphics.Canvas;
import android.graphics.Matrix;
import android.graphics.drawable.BitmapDrawable;
import android.os.Bundle;
import android.view.View;

public class IaiaiActivity extends Activity {

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

        this.setContentView(new MyView(this));
    }

    public class MyView extends View {
        private Bitmap bmp;
        private Matrix matrix = new Matrix();

        public MyView(Context context) {
            super(context);
            initialize();
        }

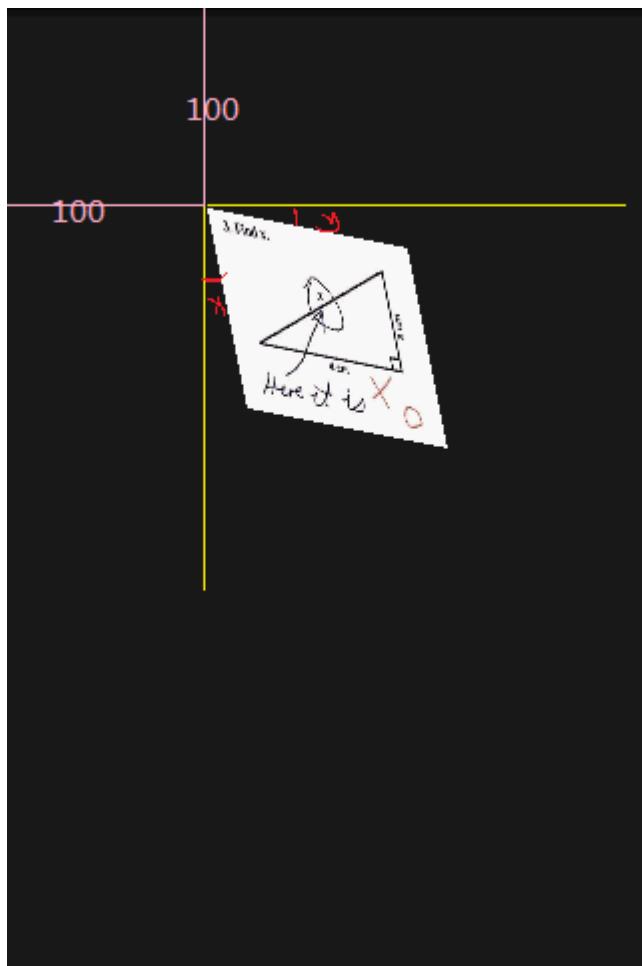
        private void initialize() {
            bmp = ((BitmapDrawable) getResources().getDrawable(R.drawable.icon))
                .getBitmap();
            // 首先，将缩放为100*100。这里scale的参数是比例。有一点要注意，如果直接用100
            matrix.setScale(100f / bmp.getWidth(), 100f / bmp.getHeight());
            // 平移到(100, 100)处
            matrix.postTranslate(100, 100);
            // 倾斜x和y轴，以(100, 100)为中心。
            matrix.postSkew(0.2f, 0.2f, 100, 100);
        }

        @Override
        protected void onDraw(Canvas canvas) {
```

```
        canvas.drawBitmap(bmp, matrix, null);
    }
}

}
```

运行结果:



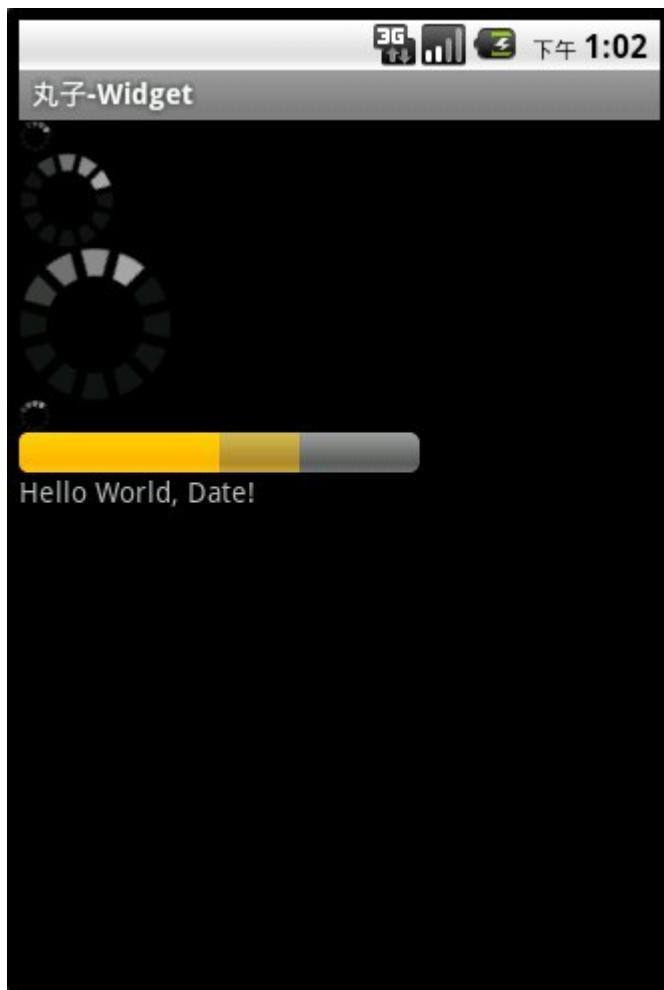
红色的x和y表示倾斜的角度，下面是x，上面是y。看到了没，Matrix就这么简单。

1.36 Android 多式样ProgressBar

发表时间: 2011-06-23 关键字: Android, ITeye, Java, XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <ProgressBar android:id="@+id/waiting1"
        style="?android:attr/progressBarStyleSmall" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_gravity="center_vertical">
    </ProgressBar>
    <ProgressBar android:id="@+id/waiting2"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_gravity="center_vertical">
    </ProgressBar>
    <ProgressBar android:id="@+id/waiting3"
        style="?android:attr/progressBarStyleLarge" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_gravity="center_vertical">
    </ProgressBar>
    <ProgressBar android:id="@+id/waiting4"
        style="?android:attr/progressBarStyleSmallTitle" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_gravity="center_vertical">
    </ProgressBar>
    <ProgressBar android:id="@+id/loading1"
        android:layout_width="200dp" android:layout_height="wrap_content"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_gravity="center_vertical" android:max="100"
        android:progress="50" android:secondaryProgress="70">
    </ProgressBar>
    <TextView android:id="@+id/txt" android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:text="@string/hello" />
</LinearLayout>
```

运行结果:



1.37 Android 边框圆角

发表时间: 2011-06-23 关键字: Android, Java, XML, ITeye

设置边框圆角可以在drawable-mdpi目录里定义一个xml:

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#000000" />
    <corners android:topLeftRadius="10dp"
              android:topRightRadius="10dp"
              android:bottomRightRadius="10dp"
              android:bottomLeftRadius="10dp"/>
</shape>
```

解释 : solid的表示填充颜色,为了简单 , 这里用的是黑色。

而corners则是表示圆角 , 注意的是这里bottomRightRadius是左下角而不是右下角 , bottomLeftRadius右下角。

当然上面的效果也可以像下面一样设置,如下:

```
<corners android:radius="5dp" />
```

如果想引用这个xml , 只需要@drawable/corners_bg.xml即可:

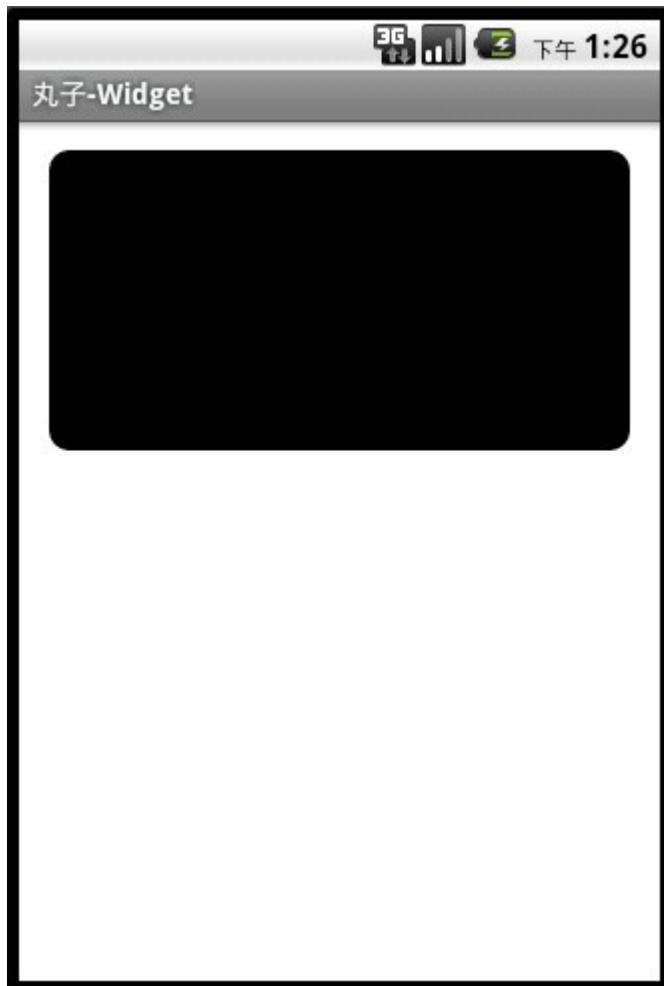
```
android:background="@drawable/corners_bg"
```

main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent" android:background="#FFFFFF">
    <RelativeLayout android:id="@+id/login_div"
        android:layout_width="fill_parent" android:layout_height="150dip"
        android:padding="15dip" android:layout_margin="15dip">
```

```
    android:background="@drawable/corners_bg">
</RelativeLayout>
</LinearLayout>
```

运行结果:



1.38 Android 背景渐变色

发表时间: 2011-06-23 关键字: Android, XML, ITeye, Java

设置背景色可以通过在drawable-mdpi/background_login.xml里定义一个xml,如下：

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <gradient android:startColor="#FFF" android:endColor="#000"
              android:angle="45" />
</shape>
```

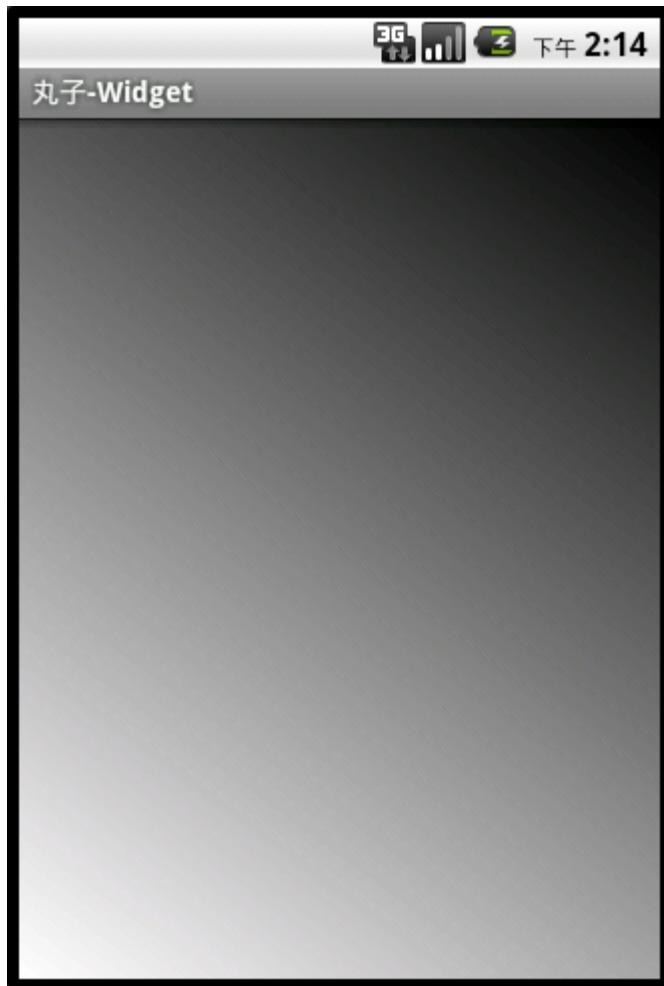
shape是用来定义形状的，gradient定义该形状里面为渐变色填充，startColor起始颜色，endColor结束颜色，angle表示方向角度。当angle=0时，渐变色是从左向右。然后逆时针方向转，当angle=90时为从下往上。

main.xml:

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

</LinearLayout>
```

运行结果:



1.39 Android Widget开发

发表时间: 2011-06-24 关键字: Android, ITeye, QQ, Java, XML

把AppWidget添加到桌面后，点击AppWidget后AppWidget文本会轮回改变

main.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:id="@+id/layout" android:layout_height="fill_parent">
    <Button android:text="@string/login" android:id="@+id/TextView01"
        android:clickable="true" android:layout_width="wrap_content"
        android:layout_height="wrap_content"></Button>
</LinearLayout>
```

res/xml/widget.xml布局文件:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- AppWidgetProviderInfo: 描述AppWidget的大小、更新频率和初始界面等信息，以XML文件形式存在于应用的r
    注意：SDK1.5之后此android:updatePeriodMillis就失效了，要自己创建service更新 --&gt;
&lt;appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
    android:minWidth="60dp" android:minHeight="30dp"
    android:updatePeriodMillis="86400000" android:initialLayout="@layout/main"&gt;
&lt;/appwidget-provider&gt;</pre>
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.iaiai.activity" android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />
    <application android:icon="@drawable/icon" android:label="@string/app_name">
```

```
<!-- <activity android:name="IaiaiActivity" android:theme="@android:style/Theme
      android:label="@string/app_name"> <intent-filter> <action android:name=
      /> <category android:name="android.intent.category.LAUNCHER" /> </inten
      </activity> -->

<receiver android:name=".IaiaiWidget" android:label="添加桌面控件">
    <meta-data android:name="android.appwidget.provider"
        android:resource="@xml/widget"></meta-data>
    <intent-filter>
        <action android:name="com.iaiai.widget.click"></action>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE">
        </intent-filter>
    </receiver>
</application>
</manifest>
```

变量类UtilTool.java：用来控件文本改变：

```
package com.iaiai.activity;

/**
 *
 * <p>
 * Title: UtilTool.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
```

```
* Create time: 2011-6-24
* </p>
*
* @author 丸子
* @version 0.0.1
*/
public class UtilTool {

    public static boolean isChange = true;

}
```

IaiaiWidget.java类：

```
package com.iaiai.activity;

import android.app.PendingIntent;
import android.appwidget.AppWidgetManager;
import android.appwidget.AppWidgetProvider;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.util.Log;
import android.widget.RemoteViews;

/**
 *
 * <p>
 * Title: IaiaiWidget.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 */
```

```
* </p>
* <p>
* Http: iaiai.iteye.com
* </p>
* <p>
* Create time: 2011-6-24
* </p>
*
* @author 丸子
* @version 0.0.1
*/
public class IaiaiWidget extends AppWidgetProvider {

    private static final String CLICK_NAME_ACTION = "com.iaiai.widget.click";

    private static RemoteViews rv;

    @Override
    public void onReceive(Context context, Intent intent) {
        super.onReceive(context, intent);

        Log.i("*****", "onReceive");

        if (rv == null) {
            rv = new RemoteViews(context.getPackageName(), R.layout.main);
        }
        if (intent.getAction().equals(CLICK_NAME_ACTION)) {
            if (UtilTool.isChange) {
                rv.setTextViewText(R.id.TextView01, "abc");
            } else {
                rv.setTextViewText(R.id.TextView01, "123");
            }
            UtilTool.isChange = !UtilTool.isChange;
            AppWidgetManager appWidgetManger = AppWidgetManager
                    .getInstance(context);
            int[] appIds = appWidgetManger.getAppWidgetIds(new ComponentName(
                    context, IaiaiWidget.class));
        }
    }
}
```

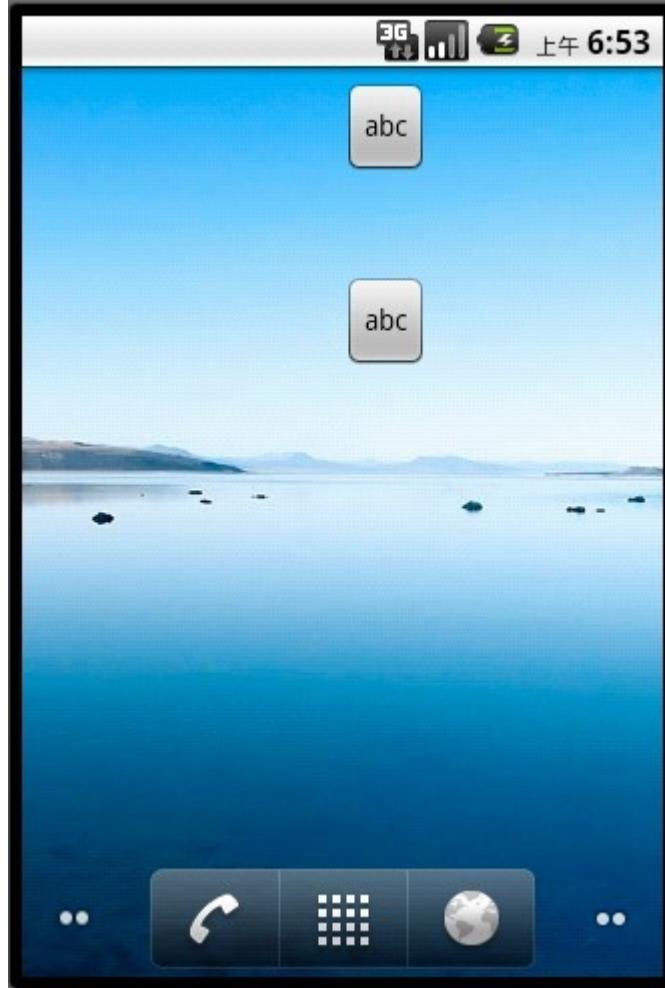
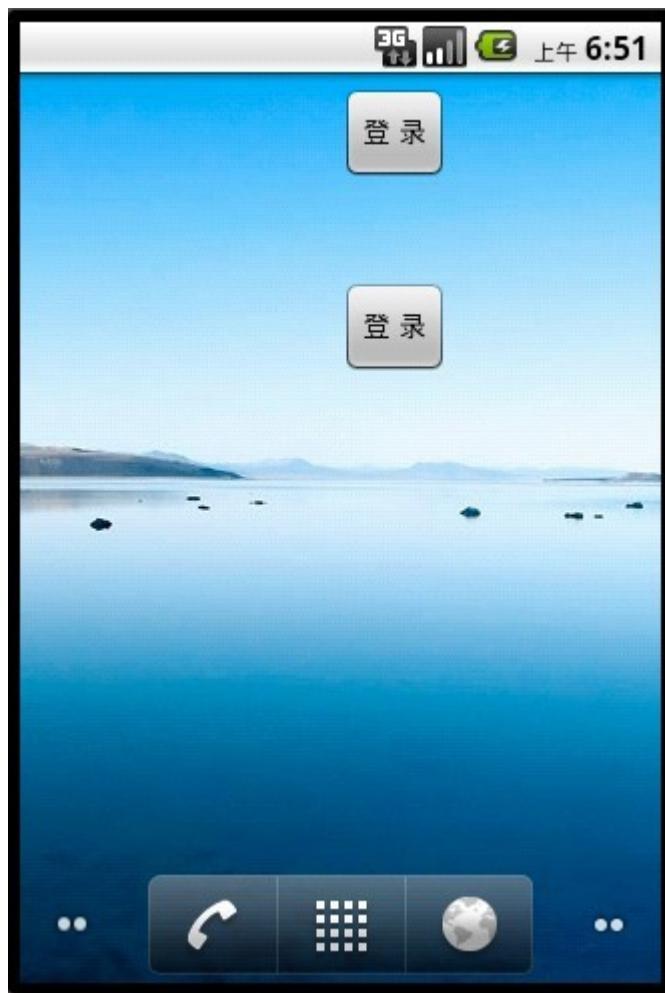
```
        appWidgetManger.updateAppWidget(appIds, rv);
    } else {
        super.onReceive(context, intent);
    }
}

@Override
public void onUpdate(Context context, AppWidgetManager appWidgetManager,
        int[] appWidgetIds) {
    Log.i("*****", "onUpdate");
    final int N = appWidgetIds.length;
    for (int i = 0; i < N; i++) {
        int appWidgetId = appWidgetIds[i];
        updateAppWidget(context, appWidgetManager, appWidgetId);
    }
}

private void updateAppWidget(Context context,
        AppWidgetManager appWidgetManger, int appWidgetId) {
    rv = new RemoteViews(context.getPackageName(), R.layout.main);
    Intent intentClick = new Intent();
    intentClick.setAction(CLICK_NAME_ACTION);
    PendingIntent pendingIntent = PendingIntent.getBroadcast(context, 0,
            intentClick, 0);
    rv.setOnClickPendingIntent(R.id.TextView01, pendingIntent);
    appWidgetManger.updateAppWidget(appWidgetId, rv);
}

}
```

运行结果:



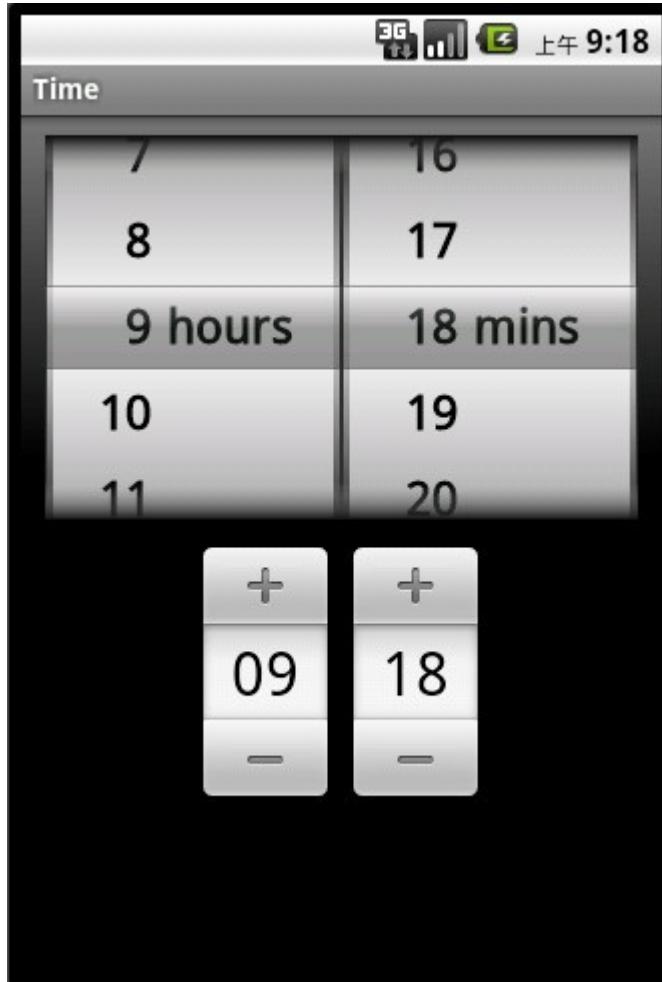
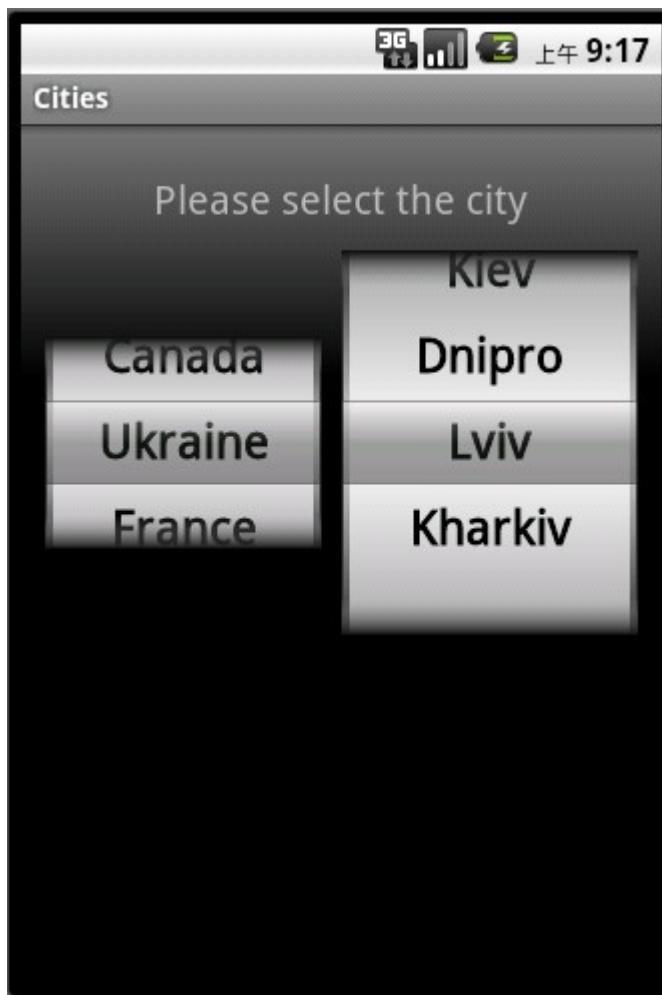
1.40 Android实现iphone风格的picker

发表时间: 2011-06-25 关键字: Android, ITeye

这是别人发的，我整理了一下。

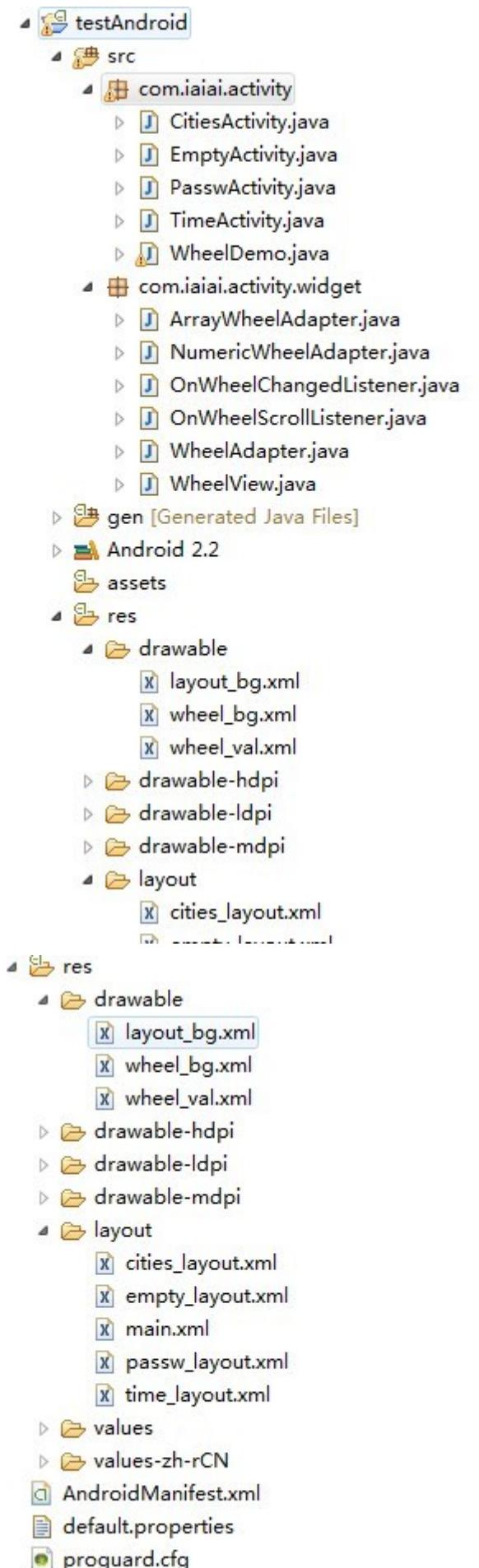
先来看下效果:





一个有意思的东西，iphone上的大滚轮，在android上也有实现了。

目录结构如下：



具体代码请下载附件里的文件。

附件下载:

- testAndroid.rar (122.9 KB)
- dl.iteye.com/topics/download/5068ee1a-d1b2-3550-8ee6-b2fc5345d243

1.41 Android 控件之SeekBar

发表时间: 2011-06-26 关键字: Android, ITeye, QQ, OS, XML

拖动条类似进度条,不同的是用户可以控制,比如,应用程序中用户可以对音效进行控制,这就可以使用拖动条来实现。由于拖动条可以被用户控制,所以需要对其进行事件监听,这就需要实现SeekBar.OnSeekBarChangeListener接口。在SeekBar中需要监听3个事件,分别是:数值的改变(onProgressChanged)、开始拖动(onStartTrackingTouch)、停止拖动(onStopTrackingTouch)。在onProgressChanged中我们可以得到当前数值的大小。我们先看看运行效果吧。



main.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <SeekBar android:id="@+id/seek" android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:max="100"
        android:progress="50" android:secondaryProgress="75" />
</LinearLayout>
```

```
<TextView android:id="@+id/progress" android:layout_width="fill_parent"
    android:layout_height="wrap_content" />

<TextView android:id="@+id/tracking" android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
</LinearLayout>
```

IaiaiActivity.java类:

```
package com.iaiai.activity;

import android.app.Activity;
import android.os.Bundle;
import android.widgetSeekBar;
import android.widget.TextView;

/**
 *
 * <p>
 * Title: IaiaiActivity.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-6-26
 * </p>
 *
 * @author 丸子
```

```
* @version 0.0.1
*/
public class IaiaiActivity extends Activity implements
    SeekBar.OnSeekBarChangeListener {

    // 声明SeekBar 和 TextView对象
    private SeekBar mSeekBar;
    private TextView mProgressText;
    private TextView mTrackingText;

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

        setContentView(R.layout.main);

        // 取得SeekBar对象
        mSeekBar = (SeekBar) findViewById(R.id.seek);
        mSeekBar.setOnSeekBarChangeListener(this);
        mProgressText = (TextView) findViewById(R.id.progress);
        mTrackingText = (TextView) findViewById(R.id.tracking);
    }

    @Override
    public void onProgressChanged(SeekBar seekBar, int progress,
        boolean fromUser) {// 在拖动中--即值在改变
        // progress为当前数值的大小
        mProgressText.setText("当前值:" + progress);
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {// 在拖动中会调用此方法
        mTrackingText.setText("正在调节");
    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {// 停止拖动
```

```
    mTrackingText.setText("xh停止调节");  
}
```

```
}
```

1.42 Android 控件之切换卡(TabWidget)

发表时间: 2011-06-26 关键字: Android, ITeye, QQ, OS, XML

TabWidget类似于Android 中查看电话薄的界面，通过多个标签切换显示不同内容。要实现这一效果，首先要了解TabHost，它是一个用来存放多个Tab标签的容器。每一个Tab都可以对自己的布局，比如，电话薄中的Tab布局就是一个List的线性布局了。

要使用TabHost,首先需要通过getTabHost方法来获取TabHost的对象，然后通过addTab方法来向TabHost中添加 Tab。当然每个Tab在切换时都会产生一个事件，要捕捉这个事件需要设置TabActivity的事件监听setOnTabChangedListener。我们先来看看运行效果吧。



main.xml布局文件:

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

```
<TabWidget android:id="@+id/tabs"
    android:layout_width="fill_parent" android:layout_height="wrap_content">
    <FrameLayout android:id="@+id/tabcontent"
        android:layout_width="fill_parent" android:layout_height="fill_parent">
            <TextView android:id="@+id/textview1" android:layout_width="fill_parent"
                android:layout_height="fill_parent" android:text="Linux"
                android:textColor="#FF0000" />
            <TextView android:id="@+id/textview2" android:layout_width="fill_parent"
                android:layout_height="fill_parent" android:textColor="#385E0F"
                android:text="MAC" />
            <TextView android:id="@+id/textview3" android:layout_width="fill_parent"
                android:layout_height="fill_parent" android:textColor="#1E90FF"
                android:text="Window" />
        </FrameLayout>
    </LinearLayout>
</TabHost>
```

IaiaiActivity.java类:

```
package com.iaiai.activity;

import android.app.AlertDialog;
import android.app.Dialog;
import android.app.TabActivity;
import android.content.DialogInterface;
import android.graphics.Color;
import android.os.Bundle;
import android.widget.TabHost;
import android.widget.TabHost.OnTabChangeListener;

/**
 *
 * <p>
 * Title: IaiaiActivity.java
 * </p>

```

```
* <p>
* E-Mail: 176291935@qq.com
* </p>
* <p>
* QQ: 176291935
* </p>
* <p>
* Http: iaiai.iteye.com
* </p>
* <p>
* Create time: 2011-6-26
* </p>
*
* @author 丸子
* @version 0.0.1
*/
public class IaiaiActivity extends TabActivity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        // 取得TabHost对象
        TabHost xh_TabHost = getTabHost();

        /**
         * 为TabHost添加标签 新建一个newTabSpec(newTabSpec) 设置其标签和图标 ( setIndicator
         * 设置内容(setContent)
         */
        // TabSpec 是TabHost的内部类 TabHost对象的 newTabSpec()方法返回一个TabSpec对象 这个
        /*
         * 源码里边是这么写的 public TabSpec newTabSpec(String tag) { return new
         * TabSpec(tag); }
         */
        xh_TabHost.addTab(xh_TabHost
```

```
        .newTabSpec("tab_test1")
        // setIndicator()此方法用来设置标签和图表
        .setIndicator("TAB 1",
                      getResources().getDrawable(R.drawable.img1))
        // 指定内容为一个TextView --->public TabHost.TabSpec setContent
        // viewId) 此方法需要一个 viewId 作为参数
        .setContent(R.id.textview1);

xh_TabHost.addTab(xh_TabHost
        .newTabSpec("tab_test2")
        .setIndicator("TAB 2",
                      getResources().getDrawable(R.drawable.img2))
        .setContent(R.id.textview2));

xh_TabHost.addTab(xh_TabHost
        .newTabSpec("tab_test3")
        .setIndicator("TAB 3",
                      getResources().getDrawable(R.drawable.img3))
        .setContent(R.id.textview3));

// 设置TabHost的背景颜色
xh_TabHost.setBackgroundColor(Color.argb(150, 22, 70, 150));

// 设置TabHost的背景图片资源
xh_TabHost.setBackgroundDrawable(R.drawable.bg2);

// 设置当前显示哪一个标签 我的理解就是当你第一次启动程序默认显示那个标签 这里是指定的
xh_TabHost.setCurrentTab(0);

// 标签切换事件处理 , setOnTabChangedListener 注意是标签切换事件不是点击事件
// 就是从一个标签切换到另外一个标签会触发的事件
xh_TabHost.setOnTabChangedListener(new OnTabChangeListener() {
    @Override
    public void onTabChanged(String tabId) {
        // 定义一个弹出式的对话框
        Dialog dialog = new AlertDialog.Builder(IaiaiActivity.this)
            .setTitle("提示")
    }
})
```

```
        .setMessage("当前选中了：" + tabId + "标签")
        .setPositiveButton("确定",
            new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    // 取消对话框
                    dialog.cancel();
                }
            })
        .create());// 创建出一个“确定”
    // 启动此对话框并且显示在屏幕上
    dialog.show();
}
});
}

}
```

附件下载:

- testAndroid.rar (1.1 MB)
- dl.iteye.com/topics/download/59145c2c-4a90-3ed7-b27f-59945ff55041

1.43 Android 标题栏(titlebar)显示进度条

发表时间: 2011-06-26 关键字: Android, ITeye

在后台线程中执行各种操作 (网络连接、大数据存储) 的时候 , 我们希望让客户能看到后台有操作在进行 , 那么既能有效的提示用户 , 又不占用当前操作空间 , 最好的方法就是在标题栏有个进度条。

实现起来很简单 :

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    requestWindowFeature(Window.FEATURE_INDETERMINATE_PROGRESS); //先给Activity注册界面  
    setContentView(R.layout.main);  
    setProgressBarIndeterminateVisibility(true); //在需要显示进度条的时候调用这个方法  
    setProgressBarIndeterminateVisibility(false); //在不需要显示进度条的时候调用这个方法  
}
```

效果如下红色线条上方所示 :



1.44 Android 控件之Notification

发表时间: 2011-06-27 关键字: Android, ITeye, Java, QQ, OS

引用

最近一直在研究android，并一边研究一边做应用。其中遇到了把程序通知常驻在Notification栏，并且不能被clear掉的问题。虽然notify()的第一个参数可以写死并clear掉，但这个值我并不想写死，但是这个值如果是随机生成一个数怎么传给Activity，用Intent当中的Bundle是可以传值过去，但是一个Notification消息是没问题，一但有多个消息，这个值就不管用了，大家有什么好的方法？

main.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Button android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:id="@+id/button"
        android:text="显示消息提示" />
</LinearLayout>
```

notification_layout.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" android:layout_height="wrap_content">
    <EditText android:text="启动Notification" android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:editable="false"
        android:cursorVisible="false" android:layout_gravity="center_horizontal" />
</LinearLayout>
```

IaiaiActivity.java类:

```
package com.iaiai.activity;

import java.util.Date;
import java.util.Random;

import android.app.Activity;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

/**
 *
 * <p>
 * Title: IaiaiActivity.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-6-26
 * </p>
 *
 * @author 丸子
 * @version 0.0.1
 */
public class IaiaiActivity extends Activity {
```

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    Button button = (Button) findViewById(R.id.button);
    View.OnClickListener listener = new View.OnClickListener() {

        @Override
        public void onClick(View view) {
            PendingIntent pending = PendingIntent.getActivity(
                IaiaiActivity.this, 0, new Intent(IaiaiActivity.this,
                    NotificationActivity.class), 0);

            Notification noticed = new Notification();
            noticed.icon = R.drawable.icon;
            noticed.tickerText = "状态栏通知";

            // noticed.defaults = Notification.DEFAULT_SOUND; // 使用默认的声音
            // noticed.defaults = Notification.DEFAULT_VIBRATE; // 使用默认的震动
            // noticed.defaults = Notification.DEFAULT_LIGHTS; // 使用默认的闪光灯
            // noticed.defaults = Notification.DEFAULT_ALL; // 所有的都使用
            noticed.defaults = Notification.DEFAULT_SOUND;

            //设置音乐
            // noticed.sound = Uri.parse("file:///sdcard/notification/ringtone.mp3");
            // noticed.sound = Uri.withAppendedPath(Audio.Media.INTERNAL_CONTENT_URI,
            //     "ringtone");

            noticed.setLatestEventInfo(IaiaiActivity.this, "这是一个状态栏通知",
                "查看", pending);
            NotificationManager noticedManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
            noticedManager.notify(0, noticed);
        }
    };
    button.setOnClickListener(listener);
}
```

```
public int getRand() {  
    Date date = new Date();  
    int year = date.getYear();  
    int month = date.getMonth();  
    int day = date.getDate();  
    return 0;  
}  
}
```

NotificationActivity.java类：

```
package com.iaiai.activity;  
  
import android.app.Activity;  
import android.app.NotificationManager;  
import android.os.Bundle;  
  
/**  
 *  
 * <p>  
 * Title: NotificationActivity.java  
 * </p>  
 * <p>  
 * E-Mail: 176291935@qq.com  
 * </p>  
 * <p>  
 * QQ: 176291935  
 * </p>  
 * <p>  
 * Http: iaiai.iteye.com  
 * </p>  
 * <p>  
 * Create time: 2011-6-26  
 * </p>
```

```
*  
* @author 丸子  
* @version 0.0.1  
*/  
public class NotificationActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.notification_layout);  
  
        Bundle bundle = getIntent().getExtras();  
  
        NotificationManager noticedManager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);  
        noticedManager.cancel(0);  
    }  
}
```

运行结果:



1.45 Android 控件之Service并开机自动启动

发表时间: 2011-06-27 关键字: Android, QQ, Java, ITeye, OS

怎样让一个Service开机自动启动

AndroidManifest.xml配置文件:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.iaiai.activity" android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".IaiaiActivity" android:theme="@android:style/Theme.Dialog">
            <intent-filter>
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <receiver android:name=".IaiaiReceiver" android:permission="android.permission.RECEIVE_BOOT_COMPLETED">
            <intent-filter>
                <action android:name="android.intent.action.BOOT_COMPLETED" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </receiver>
        <service android:name=".IaiaiService" android:enabled="true" />
    </application>

    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
</manifest>
```

main.xml布局文件:

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

```
    android:layout_height="fill_parent">
    <Button android:layout_width="fill_parent"
            android:layout_height="wrap_content" android:id="@+id/button"
            android:text="开机自动启动" />
</LinearLayout>
```

IaiaiActivity.java类:

```
package com.iaiai.activity;

import android.app.Activity;
import android.content.ComponentName;
import android.content.Intent;
import android.content.ServiceConnection;
import android.os.Bundle;
import android.os.IBinder;

/**
 *
 * <p>
 * Title: IaiaiActivity.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-6-26
 * </p>
 *
```

```
* @author 丸子
* @version 0.0.1
*/
public class IaiaiActivity extends Activity {

    private IaiaiService iaiaiService;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Intent intent = new Intent(this, IaiaiService.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

        startService(intent);
        stopService(intent);
    }

    private ServiceConnection serviecConnection = new ServiceConnection() {
        public void onServiceConnected(ComponentName classname, IBinder service) {
            iaiaiService = ((IaiaiService.IaiaiBinder) (service)).getService();
        }

        public void onServiceDisconnected(ComponentName classname) {
            iaiaiService = null;
        }
    };
}

}
```

IaiaiService.java类:

```
package com.iaiai.activity;
```

```
import android.app.Service;
import android.content.Intent;
import android.os.Binder;
import android.os.IBinder;
import android.util.Log;

/**
 *
 * <p>
 * Title: IaiaiService.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-6-27
 * </p>
 *
 * @author 丸子
 * @version 0.0.1
 */
public class IaiaiService extends Service {

    @Override
    public IBinder onBind(Intent intent) {
        Log.i("*****", "onBind");
        return null;
    }

    @Override
    public void onCreate() {
```

```
        Log.i("*****", "onCreate");
    }

@Override
public void onDestroy() {
    Log.i("*****", "onDestroy");
}

@Override
public void onStart(Intent intent, int startId) {
    Log.i("*****", "onStart");
}

@Override
public boolean onUnbind(Intent intent) {
    Log.i("*****", "onUnbind");
    return super.onUnbind(intent);
}

public class IaiaiBinder extends Binder {
    public IaiaiService getService() {
        return IaiaiService.this;
    }
}

}
```

IaiaiReceiver.java类:

```
package com.iaiai.activity;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
```

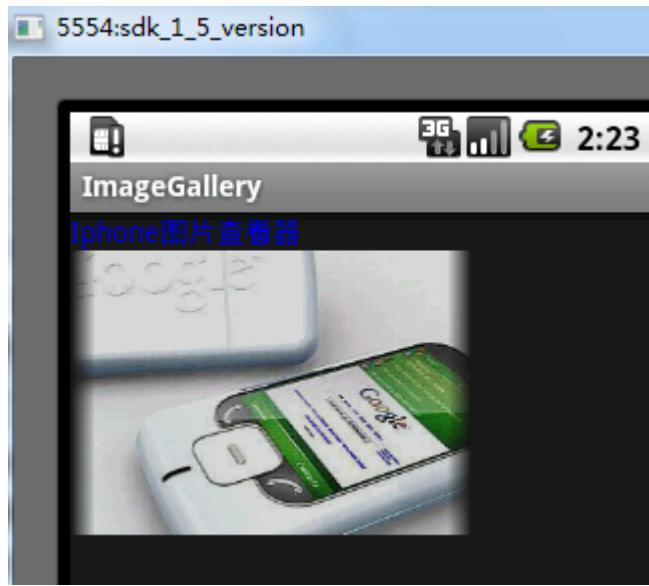
```
/**  
 *  
 * <p>  
 * Title: IaiaiReceiver.java  
 * </p>  
 * <p>  
 * E-Mail: 176291935@qq.com  
 * </p>  
 * <p>  
 * QQ: 176291935  
 * </p>  
 * <p>  
 * Http: iaiai.iteye.com  
 * </p>  
 * <p>  
 * Create time: 2011-6-27  
 * </p>  
 *  
 * @author 丸子  
 * @version 0.0.1  
 */  
  
public class IaiaiReceiver extends BroadcastReceiver {  
  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        Intent serviceIntent = new Intent(context, IaiaiService.class);  
        context.startService(serviceIntent);  
  
        Intent activityIntent = new Intent(context, IaiaiActivity.class);  
        activityIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);  
        context.startActivity(activityIntent);  
    }  
}
```

运行结果:



1.46 Android 拖动换图特效

发表时间: 2011-06-27 关键字: Android, ITeye, QQ, OS, C



曾经被iPhone用手指在屏幕上滑来滑去拖动图片的操作方式吸引吗？在Android里头，这并不是什么难事。

IaiaiActivity.java类:

```
package com.iaiai.activity;

import android.app.Activity;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.ServiceConnection;
import android.os.Bundle;
import android.os.IBinder;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.Gallery;
import android.widget.ImageView;

/**
 * 
 */
```

```
*  
* <p>  
* Title: IaiaiActivity.java  
* </p>  
* <p>  
* E-Mail: 176291935@qq.com  
* </p>  
* <p>  
* QQ: 176291935  
* </p>  
* <p>  
* Http: iaiai.iteye.com  
* </p>  
* <p>  
* Create time: 2011-6-26  
* </p>  
*  
* @author 丸子  
* @version 0.0.1  
*/  
  
public class IaiaiActivity extends Activity {  
  
    private IaiaiService iaiaiService;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
  
        Gallery mGallery = (Gallery) findViewById(R.id.Gallery01);  
        mGallery.setAdapter(new ImageApdater(this));  
    }  
  
    public class ImageApdater extends BaseAdapter {  
  
        // 类成员myContext为context父类  
        private Context myContext;
```

```
private int[] myImageIds = { R.drawable.a, R.drawable.b, R.drawable.c,
                            R.drawable.d, R.drawable.e };

// 构造函数，有一个参数，即要存储的Context
public ImageApdater(Context c) {
    // TODO Auto-generated constructor stub
    this.mContext = c;
}

// 返回所有的图片总数量
@Override
public int getCount() {
    // TODO Auto-generated method stub
    return this.myImageIds.length;
}

// 利用getItem方法，取得目前容器中图像的数组ID
@Override
public Object getItem(int position) {
    // TODO Auto-generated method stub
    return position;
}

@Override
public long getItemId(int position) {
    // TODO Auto-generated method stub
    return position;
}

// 取得目前欲显示的图像的VIEW，传入数组ID值使之读取与成像
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    // TODO Auto-generated method stub
    ImageView i = new ImageView(this.mContext);
    i.setImageResource(this.myImageIds[position]);
    i.setScaleType(ImageView.ScaleType.FIT_XY);
    // i.setLayoutParams(new Gallery.LayoutParams(120,120));
}
```

```
        return i;  
    }  
  
}  
  
}
```

main.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical" android:layout_width="fill_parent"  
    android:layout_height="fill_parent">  
    <Gallery android:id="@+id/Gallery01" android:layout_width="wrap_content" android:layout  
    </LinearLayout>
```

1.47 Android 使用Gallery实现Tab

发表时间: 2011-06-30

main.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Gallery android:id="@+id/gallery" android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:layout_marginTop="30dp"
        android:unselectedAlpha="1" android:spacing="1dip" />
</LinearLayout>
```

values/attrs.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <declare-styleable name="Gallery">
        <attr name="android:galleryItemBackground" />
    </declare-styleable>
</resources>
```

values/strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, Date!</string>
    <string name="app_name">丸子-Widget</string>
</resources>
```

drawable/tab_button_select.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <gradient android:startColor="#FF1B1B1B" android:endColor="#FF969696"
        android:angle="90.0">
    </gradient>
</shape>
```

drawable/tab_button_unselect.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <gradient android:startColor="#FF000000" android:endColor="#FF474747"
        android:angle="90.0">
    </gradient>
</shape>
```

IaiaiActivity.java类:

```
package com.iaiai.activity;

import java.util.Arrays;
import java.util.Collections;
import java.util.List;

import android.app.Activity;
import android.content.Context;
import android.content.res.TypedArray;
import android.graphics.Color;
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
```

```
import android.widget.AdapterView.OnItemClickListener;
import android.widget.BaseAdapter;
import android.widget.Gallery;
import android.widget.TextView;

/**
 *
 * <p>
 * Title: IaiaiActivity.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-6-26
 * </p>
 *
 * @author 丸子
 * @version 0.0.1
 */
public class IaiaiActivity extends Activity {

    private Gallery gallery;
    private TabAdapter textAdapter;

    private static final String[] PROGRAM_NAMES = { "中央一台", "中央二台", "中央三台",
        "中央四台", "中央五台", "中央六台", "中央七台", "中央八台", };

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

```
        setContentView(R.layout.main);

        gallery = (Gallery) findViewById(R.id.gallery);
        textAdapter = new TabAdapter(this, Arrays.asList(PROGRAM_NAMES));
        gallery.setAdapter(textAdapter);
        gallery.setOnItemClickListener(new OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView<?> parent, View view,
                    int position, long id) {
                TabAdapter adapter = (TabAdapter) parent.getAdapter();
                adapter.setSelectedPos(position);
            }
        });
    }

    public class TabAdapter extends BaseAdapter {
        private Context mContext;
        private List<String> mList;
        private int mSelectedPos;

        public TabAdapter(Context context, List<String> list) {
            mContext = context;
            TypedArray a = obtainStyledAttributes(R.styleable.Gallery);
            a.recycle();
            if (list == null)
                list = Collections.emptyList();
            mList = list;
        }

        public void setSelectedPos(int pos) {
            if (pos != mSelectedPos) {
                mSelectedPos = pos;
                notifyDataSetChanged();
            }
        }
    }
}
```

```
public int getSelectedPos() {
    return mSelectedPos;
}

public int getCount() {
    return mList.size();
}

public Object getItem(int position) {
    return mList.get(position);
}

public long getItemId(int position) {
    return position;
}

public View getView(int position, View convertView, ViewGroup parent) {
    TextView text = null;
    if (convertView == null) {
        text = new TextView(mContext);
    } else {
        text = (TextView) convertView;
    }

    text.setTextColor(Color.WHITE);
    text.setText(mList.get(position));

    text.setLayoutParams(new Gallery.LayoutParams(102, 40));
    text.setGravity(Gravity.CENTER);

    if (position == mSelectedPos)
        text.setBackgroundResource(R.drawable.tab_button_select);
    else
        text.setBackgroundResource(R.drawable.tab_button_unselect);

    return text;
}
```

```
    }  
}  
  
}
```

运行结果:



1.48 Android ListView的分段显示、分页显示(一)

发表时间: 2011-07-09

直接上代码:

IaiaiActivity.java:

```
package com.iaiai.activity;

import android.app.ListActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Gravity;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AbsListView;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.LinearLayout.LayoutParams;
import android.widget.ListView;
import android.widget.TextView;

/**
 *
 * <p>
 * Title: IaiaiActivity.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
```

```
* <p>
* Create time: 2011-6-26
* </p>
*
* @author 丸子
* @version 0.0.1
*/
public class IaiaiActivity extends ListActivity {

    private static final String TAG = "IaiaiActivity";

    private listViewAdapter adapter = new listViewAdapter();
    /**
     * 设置布局显示为目标有多大就多大
     */
    private LayoutParams WLayoutParams = new LinearLayout.LayoutParams(
            LinearLayout.LayoutParams.WRAP_CONTENT,
            LinearLayout.LayoutParams.WRAP_CONTENT);
    /**
     * 设置布局显示目标最大化
     */
    private LayoutParams FLayoutParams = new LinearLayout.LayoutParams(
            LinearLayout.LayoutParams.FILL_PARENT,
            LinearLayout.LayoutParams.FILL_PARENT);

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Log.i(TAG, "onCreate(Bundle savedInstanceState):");
        LinearLayout layout = new LinearLayout(this);
        layout.setOrientation(LinearLayout.HORIZONTAL);

        Button button = new Button(this);
        button.setText("点击加载下五条...\"");
        button.setGravity(Gravity.CENTER_VERTICAL);

        layout.addView(button, FLayoutParams);
        layout.setGravity(Gravity.CENTER);
    }
}
```

```
LinearLayout loadingLayout = new LinearLayout(this);
loadingLayout.addView(layout, WLayoutParams);
loadingLayout.setGravity(Gravity.CENTER);

ListView listView = getListView();

listView.addFooterView(loadingLayout);

button.setOnClickListener(new Button.OnClickListener() {

    @Override
    public void onClick(View v) {
        adapter.count += 5;
        Log.i(TAG, "setOnClickListener:" + adapter.count);
        adapter.notifyDataSetChanged();
    }
});

setListAdapter(adapter);
}

public void onScrollStateChanged(AbsListView v, int s) {
    Log.i(TAG, "onScrollStateChanged(AbsListView v, int s)");
}

class listViewAdapter extends BaseAdapter {
    int count = 10; /* starting amount */

    public int getCount() {
        return count;
    }

    public Object getItem(int pos) {
        return pos;
    }

    public long getItemId(int pos) {
        return pos;
    }
}
```

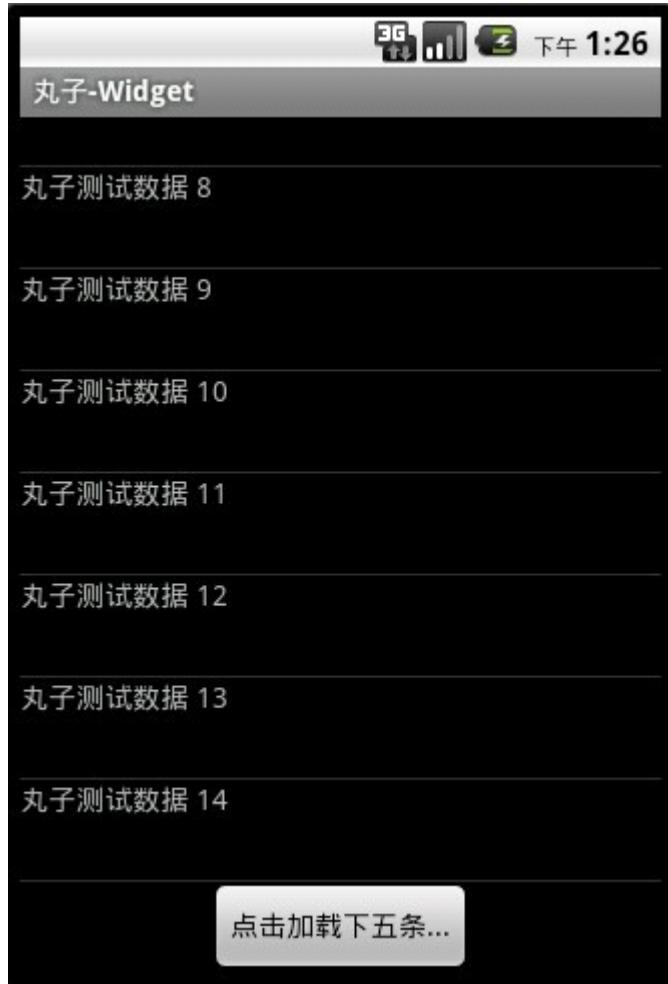
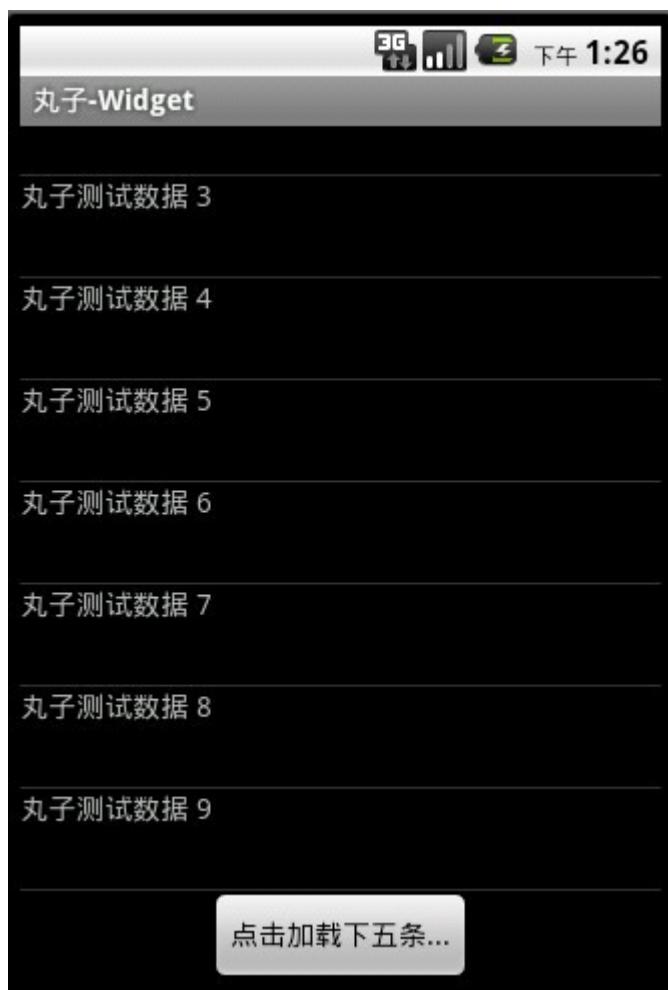
```
}

public View getView(int pos, View v, ViewGroup p) {
    TextView textView = new TextView(IaiaiActivity.this);
    textView.setHeight(50);
    textView.setText("丸子测试数据 " + pos);
    return textView;
}

}

}
```

运行结果:



1.49 Android ListView的分段显示、分页显示(二)

发表时间: 2011-07-09

直接上代码

IaiaiActivity.java:

```
package com.iaiai.activity;

import android.app.ListActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Gravity;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AbsListView;
import android.widget.AbsListView.OnScrollListener;
import android.widget.BaseAdapter;
import android.widget.LinearLayout;
import android.widget.LinearLayout.LayoutParams;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.TextView;

/**
 *
 * <p>
 * Title: IaiaiActivity.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 */
```

```
* </p>
* <p>
* Create time: 2011-6-26
* </p>
*
* @author 丸子
* @version 0.0.1
*/
public class IaiaiActivity extends ListActivity implements OnScrollListener {

    private static final String TAG = "IaiaiActivity";
    private listViewAdapter adapter = new listViewAdapter();
    private int lastItem = 0;
    /**
     * 设置布局显示为目标有多大就多大
     */
    private LayoutParams WLayoutParams = new LinearLayout.LayoutParams(
        LinearLayout.LayoutParams.WRAP_CONTENT,
        LinearLayout.LayoutParams.WRAP_CONTENT);

    /**
     * 设置布局显示目标最大化
     */
    private LayoutParams FLayoutParams = new LinearLayout.LayoutParams(
        LinearLayout.LayoutParams.FILL_PARENT,
        LinearLayout.LayoutParams.FILL_PARENT);

    private ProgressBar progressBar;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        Log.i(TAG, "onCreate(Bundle savedInstanceState)");
        // 线性布局
        LinearLayout layout = new LinearLayout(this);
        // 设置布局 水平方向
        layout.setOrientation(LinearLayout.HORIZONTAL);
        // 进度条
```

```
progressBar = new ProgressBar(this);
// 进度条显示位置
progressBar.setPadding(0, 0, 15, 0);

layout.addView(progressBar, WLayoutParams);

TextView textView = new TextView(this);
textView.setText("加载中...");
textView.setGravity(Gravity.CENTER_VERTICAL);

layout.addView(textView, FLayoutParams);
layout.setGravity(Gravity.CENTER);

LinearLayout loadingLayout = new LinearLayout(this);
loadingLayout.addView(layout, WLayoutParams);
loadingLayout.setGravity(Gravity.CENTER);

// 得到一个ListView用来显示条目
ListView listView = getListView();
// 添加到脚页显示
listView.addFooterView(loadingLayout);
//
registerForContextMenu(listView);
//
setListAdapter(adapter);
listView.setOnScrollListener(this);

}

public void onScroll(AbsListView v, int firstVisibleItem,
                     int visibleItemCount, int totalItemCount) {
    lastItem = firstVisibleItem + visibleItemCount - 1;
    System.out.println("lastItem:" + lastItem);
    Log.i(TAG, "lastItem:" + lastItem);

}

public void onScrollStateChanged(AbsListView v, int state) {
```

```
if (lastItem == adapter.count
        && state == OnScrollListener.SCROLL_STATE_IDLE) {
    adapter.count += 10;
    adapter.notifyDataSetChanged();
    Log.i(TAG, "lastItem:" + lastItem);
}
}

/**
 * 要用用于生成显示数据
 */
class listViewAdapter extends BaseAdapter {
    int count = 10;

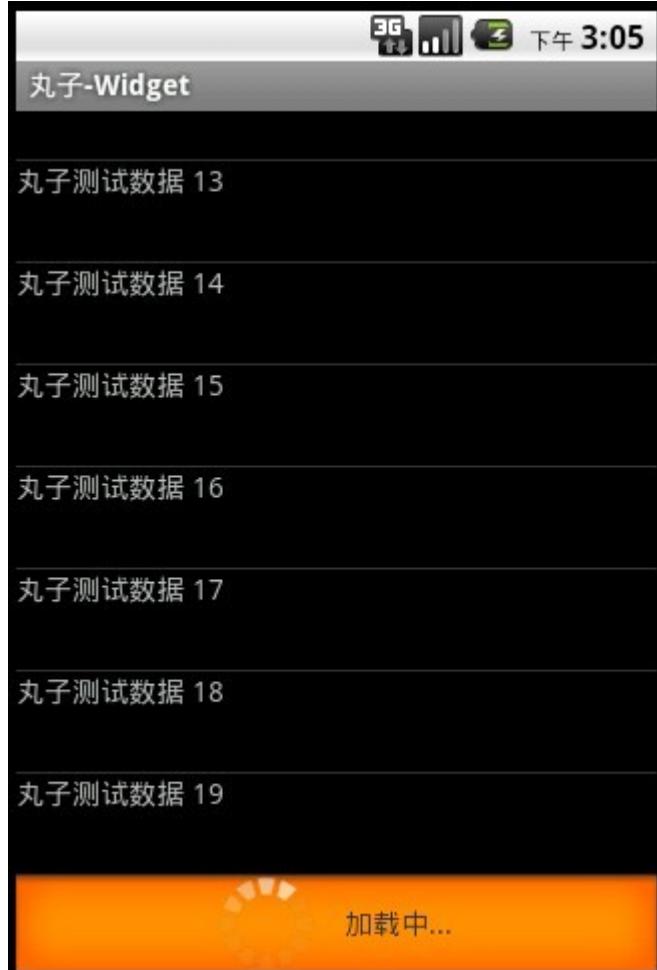
    public int getCount() {
        Log.i(TAG, "count:" + count);
        return count;
    }

    public Object getItem(int pos) {
        Log.i(TAG, "pos:" + pos);
        return pos;
    }

    public long getItemId(int pos) {
        return pos;
    }

    public View getView(int pos, View v, ViewGroup p) {
        TextView view = new TextView(IaiaiActivity.this);
        view.setText("丸子测试数据 " + pos);
        Log.i(TAG, "entry: " + pos);
        view.setHeight(50);
        return view;
    }
}
```

}



1.50 Android sqllite中日期时间问题

发表时间: 2011-07-10

时间，使用了方法：

```
public void addScore(String nickname, float score, String status){  
    SQLiteDatabase db = this.getWritableDatabase();  
  
    String sql = "INSERT INTO scores(test_datetime, nickname, score, status)"+  
        " VALUES(datetime('now', 'localtime'), ?, ?, ?)";  
    Object[] bindArgs = new Object[]{nickname, score, status};  
    try{  
        db.execSQL(sql, bindArgs);  
    }catch(SQLException e){  
        Log.e("wang", e.toString());  
    }  
}
```

测试没通过，说总是差8个小时，发现(datetime('now'))错了，应该是datetime('now','localtime')这样才记录手机的本地时间，不然记录的是手机的格林威治时间。

总结：

- 1、在使用模拟机时，一定要更改时区，有问题会及时发现。
- 2、自己测试要仔细些。

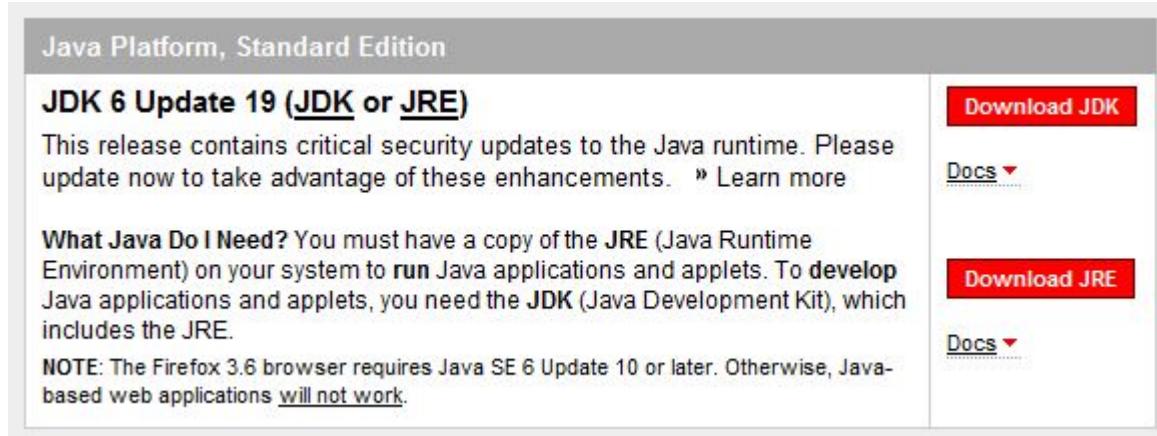
1.51 Android开发环境搭建全程演示

发表时间: 2011-07-11

一 相关下载

(1) java JDK下载:

进入该网页: <http://java.sun.com/javase/downloads/index.jsp> 如下图:



Java Platform, Standard Edition

JDK 6 Update 19 (JDK or JRE)

This release contains critical security updates to the Java runtime. Please update now to take advantage of these enhancements. [» Learn more](#)

What Java Do I Need? You must have a copy of the JRE (Java Runtime Environment) on your system to **run** Java applications and applets. To **develop** Java applications and applets, you need the **JDK** (Java Development Kit), which includes the JRE.

NOTE: The Firefox 3.6 browser requires Java SE 6 Update 10 or later. Otherwise, Java-based web applications will not work.

Download JDK

Docs

Download JRE

Docs

选择 Download JDK 只下载JDK，无需下载jre.

(2)eclipse下载

进入该网页: <http://www.eclipse.org/downloads/> 如下图:



Package	Description	Downloads	Platform
Eclipse IDE for Java EE Developers (190 MB)	Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn and others. More...	868,089	Windows 32bit Mac Carbon 32bit Mac Cocoa 32bit 64bit Linux 32bit 64bit
Eclipse IDE for Java Developers (92 MB)	The essential tools for any Java developer, including a Java IDE, a CVS client, XML Editor and Mylyn. More...	298,541	Windows 32bit Mac Carbon 32bit Mac Cocoa 32bit 64bit Linux 32bit 64bit
Eclipse IDE for C/C++ Developers (79 MB)	An IDE for C/C++ developers with Mylyn integration. More...	130,736	Windows 32bit Mac Carbon 32bit Mac Cocoa 32bit 64bit Linux 32bit 64bit
Eclipse for PHP Developers (139 MB)	Tools for PHP developers creating Web applications, including PHP Development Tools (PDT), Web Tools Platform, Mylyn and others. More...	103,015	Windows 32bit Mac Carbon 32bit Mac Cocoa 32bit 64bit Linux 32bit 64bit

我们选择第一个(即eclipse IDE for java EE Developers)

(3)下载Android SDK

说明: Android SDK两种下载版本，一种是包含具体版本的SDK的，一种是只有升级工具，而不包含具体的SDK

版本，后一种大概20多M，前一种70多M。

完全版下载 (android sdk 2.1 r01)

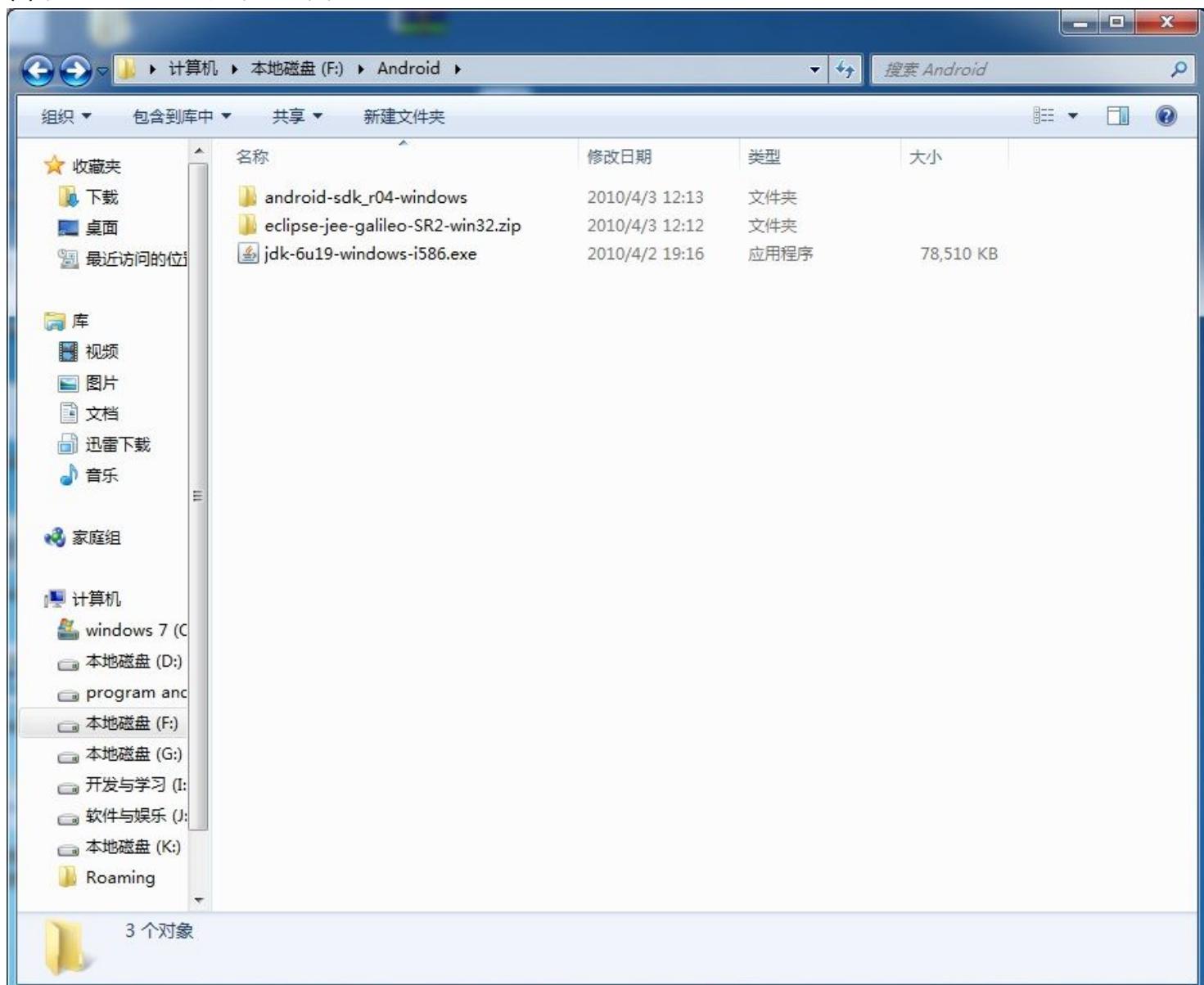
https://dl-ssl.google.com/android/repository/android-2.1_r01-windows.zip

升级版下载 (建议使用这个，本例子就是使用这个这里面不包含具体版本，想要什么版本在Eclipse里面升级就行)

<http://developer.android.com/sdk/index.html>

二 软件安装

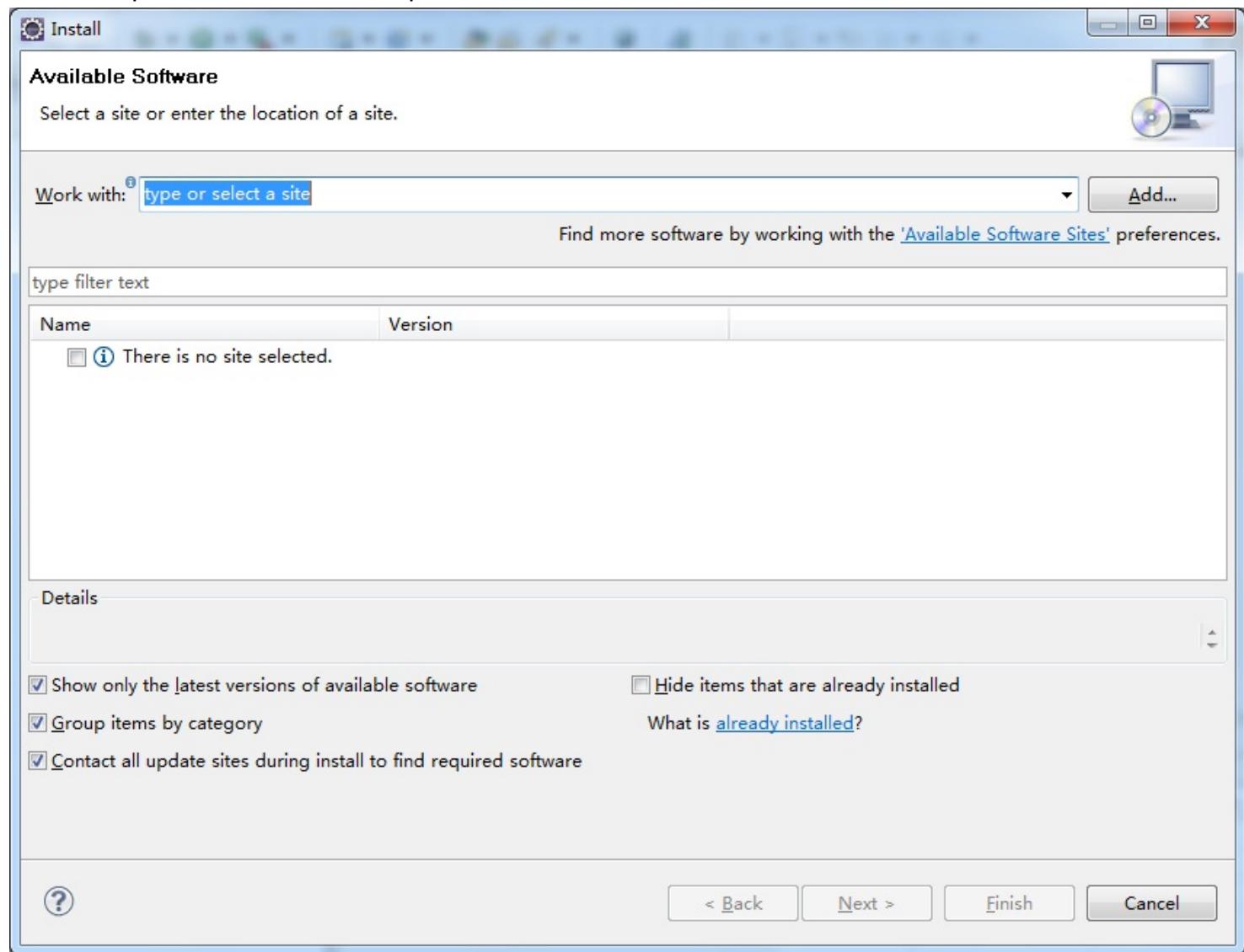
- (1)安装jdk 6u19 安装完成即可，无需配置环境变量
- (2)解压eclipse eclipse无需安装，解压后，直接打开就行
- (3)解压android sdk 这个也无需安装，解压后供后面使用
- (4)最终有三个文件夹，如下图：



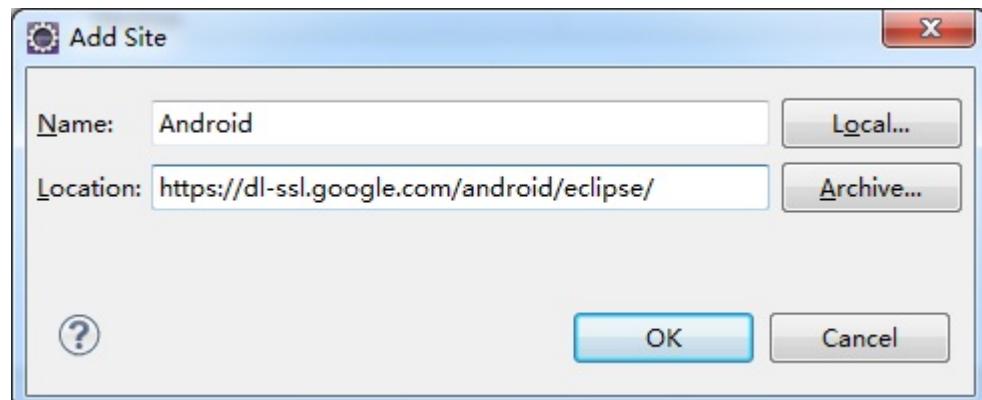
三 Eclipse配置

1 安装android 开发插件

(1) 打开Eclipse, 在菜单栏上选择 help->Install New SoftWare 出现如下界面:



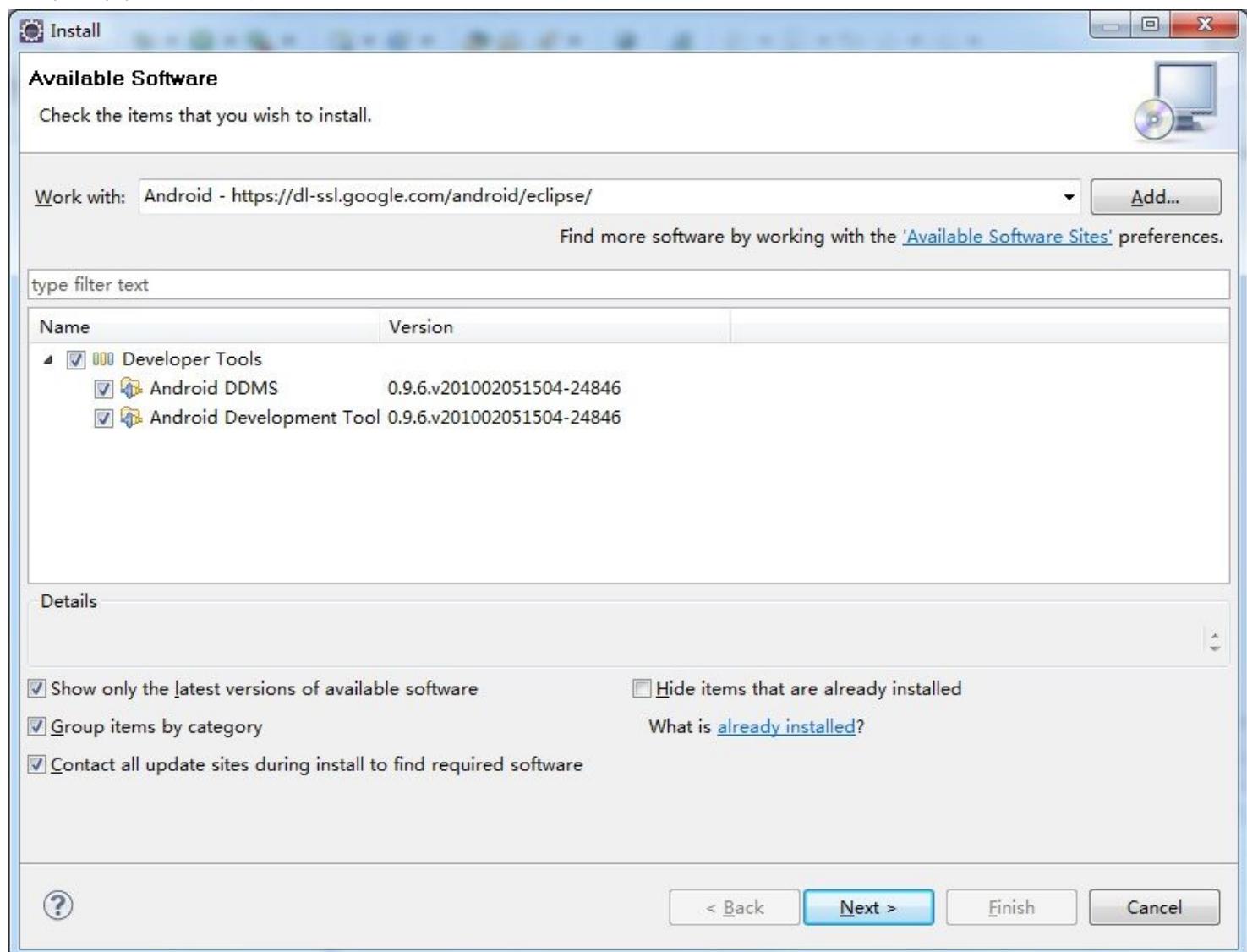
点击Add按钮,出现如下界面



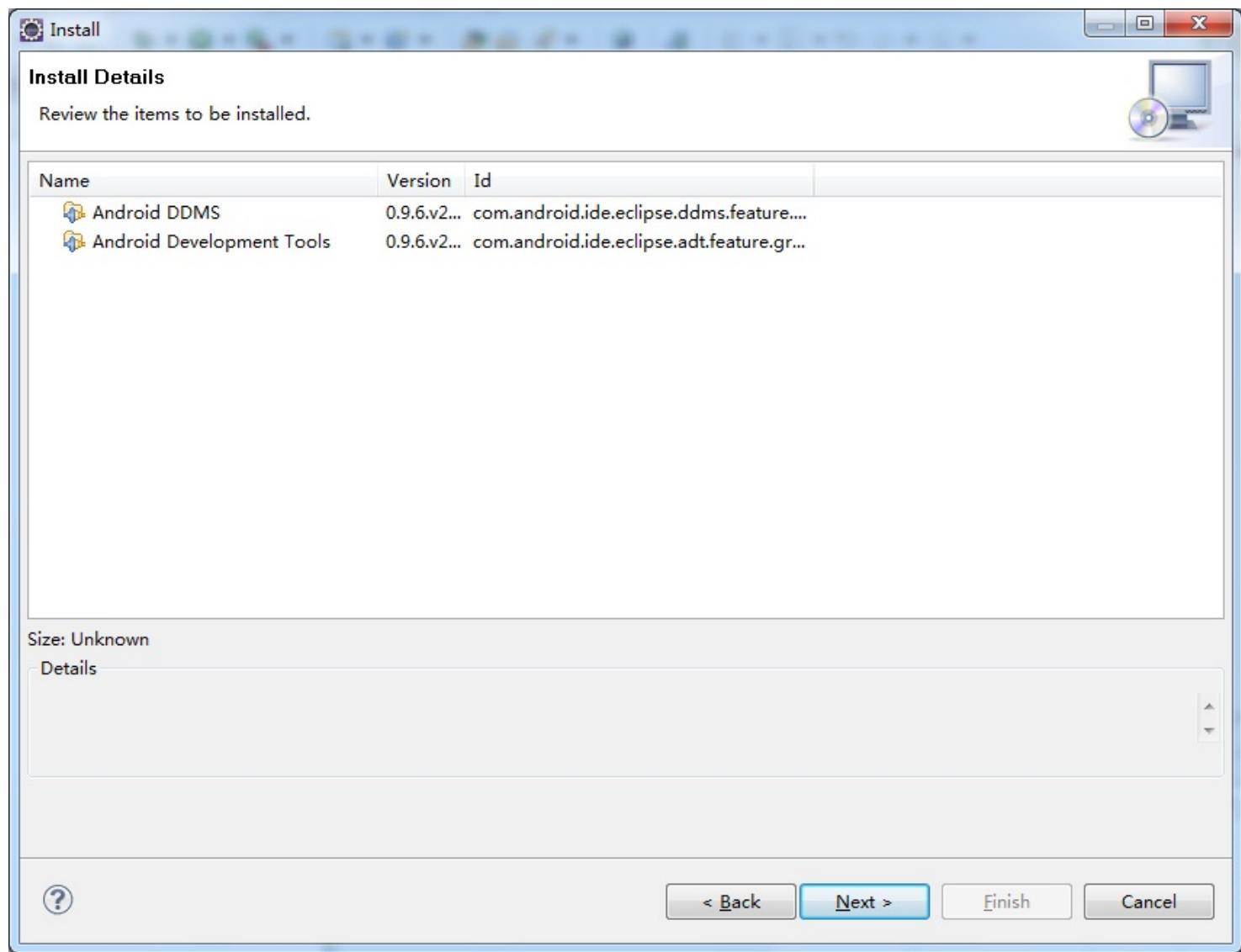
输入网址: <https://dl-ssl.google.com/android/eclipse/> (如果出错, 请将https改成http)

名称: Android (这里可以自定义)

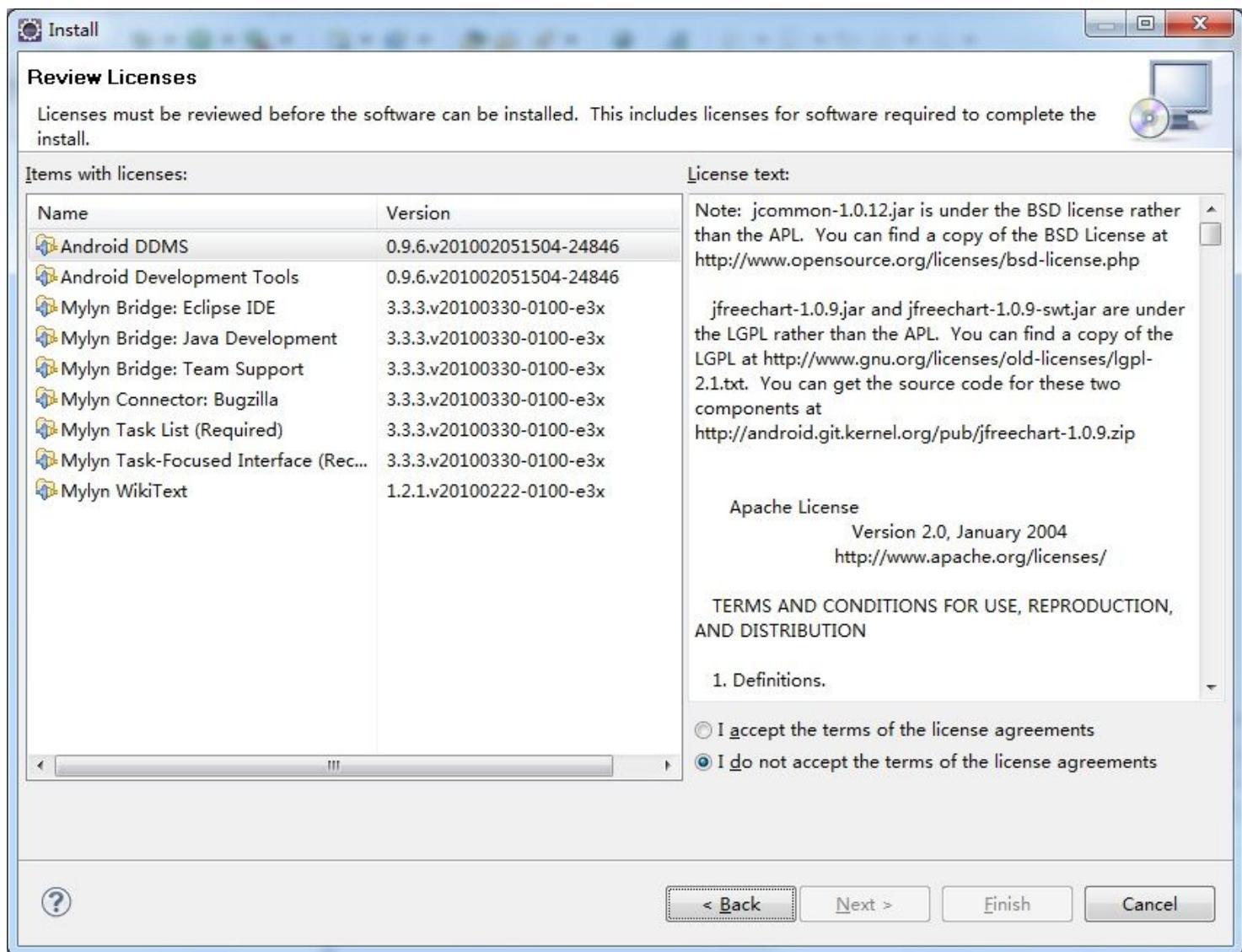
点击OK , 将出现如下界面



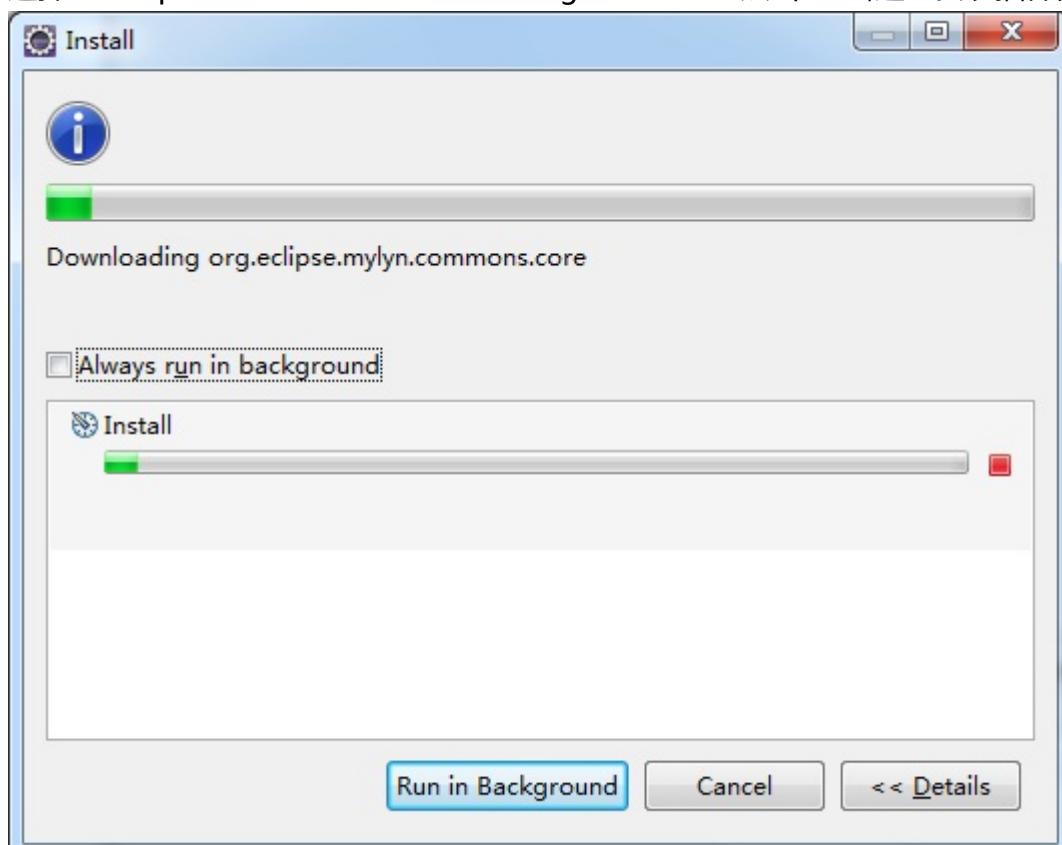
点击 Next按钮 , 出现如下界面:



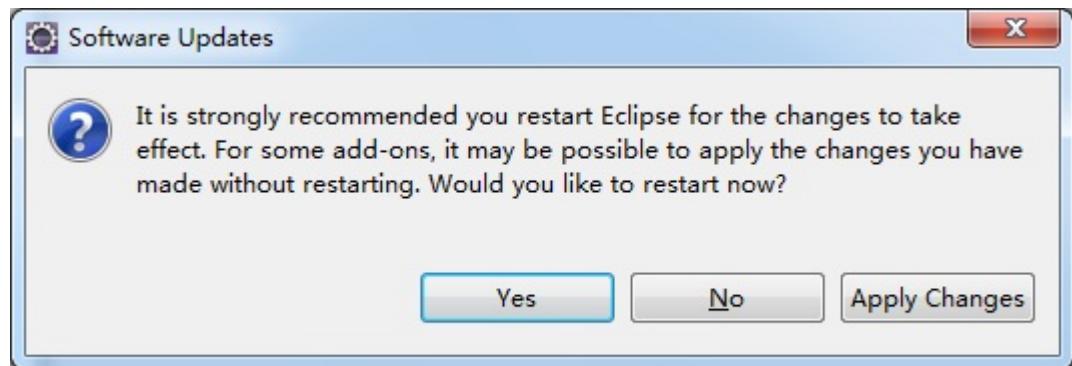
点击Next按钮，出现如下界面：



选择 I accept the terms of the license agreements 点击Next,进入安装插件界面



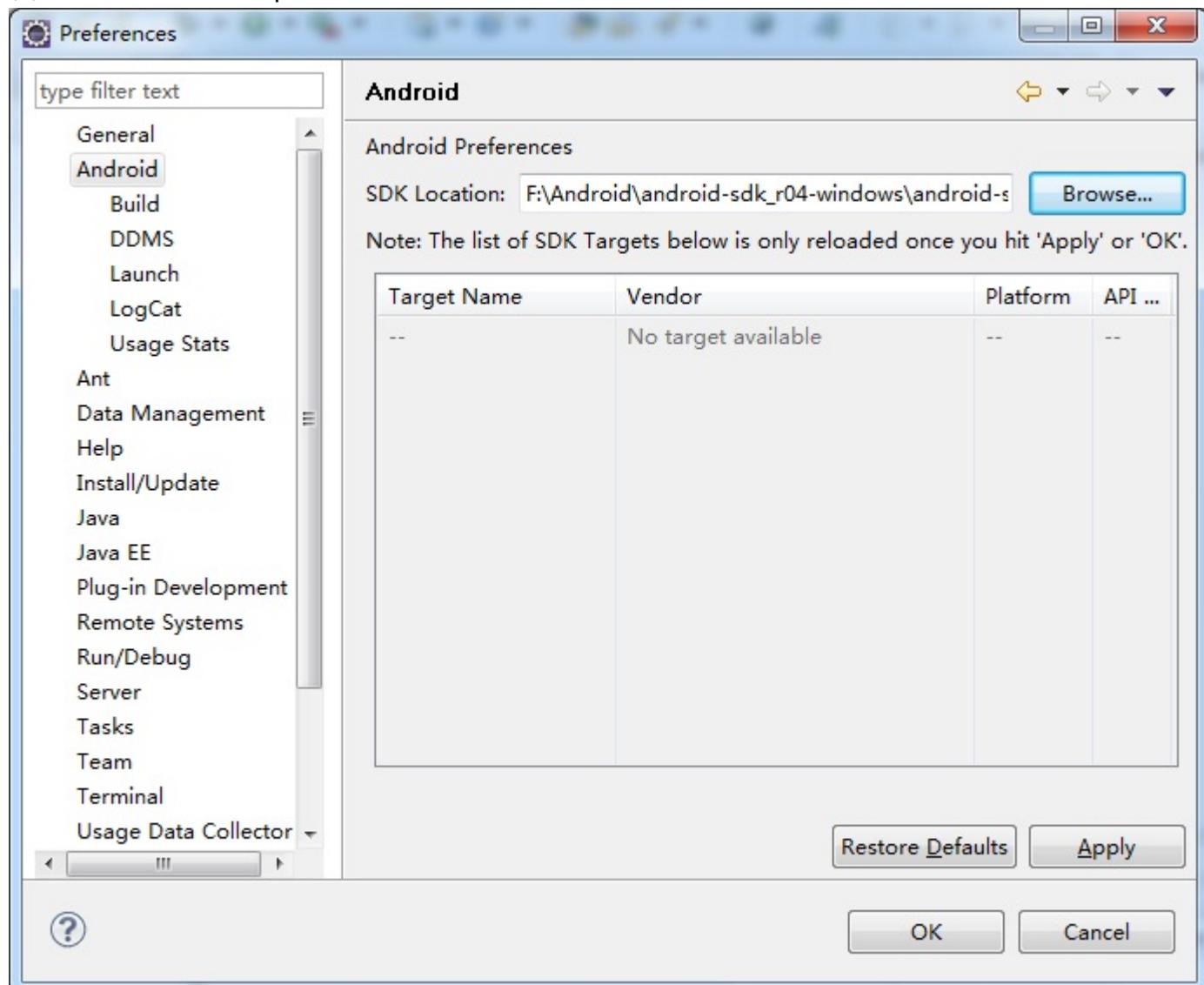
安装完成后，出现如下界面



点击Yes按钮，重启Eclipse

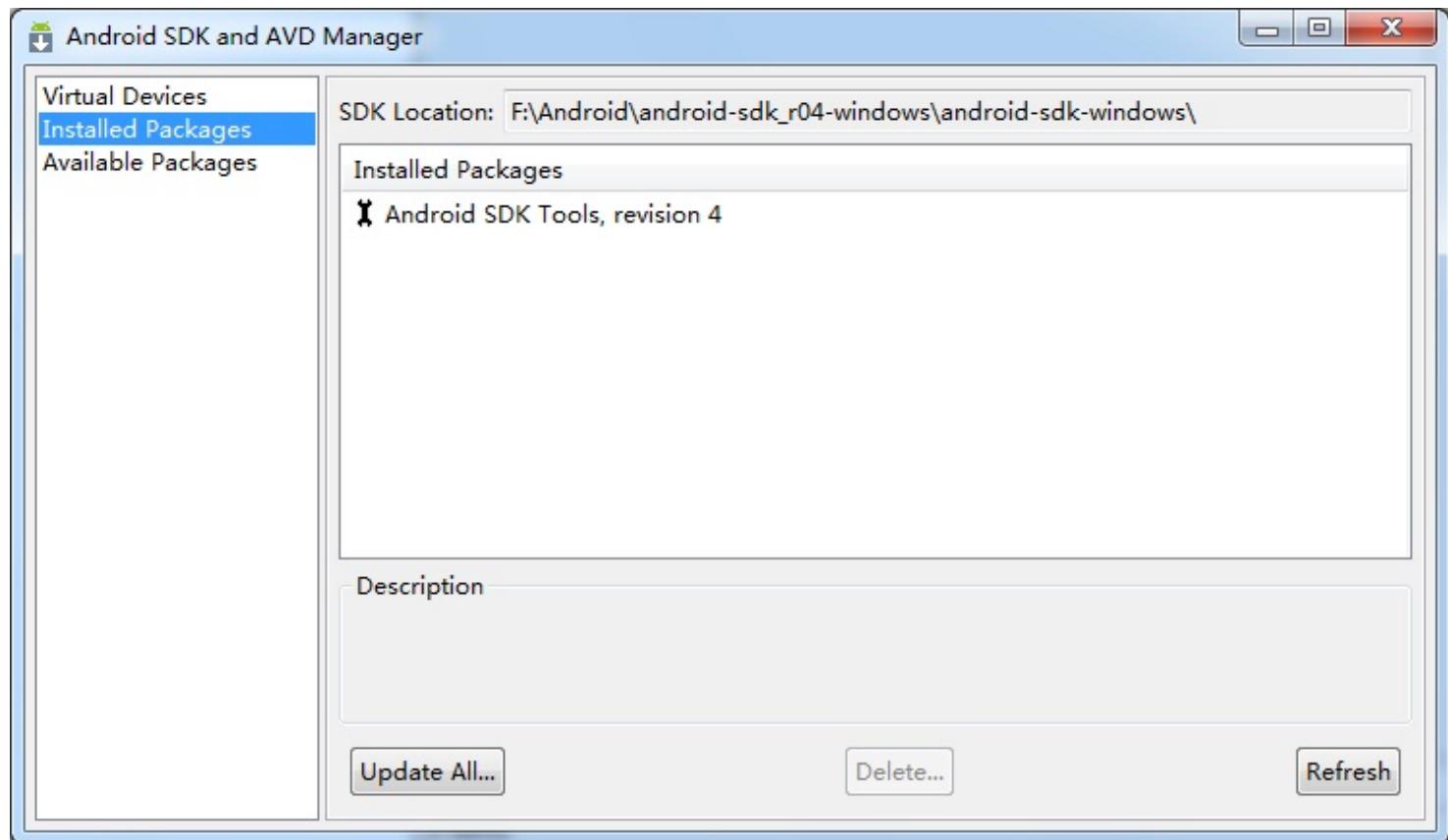
2 配置android sdk

(1)点击菜单window->preferences,进入如下界面

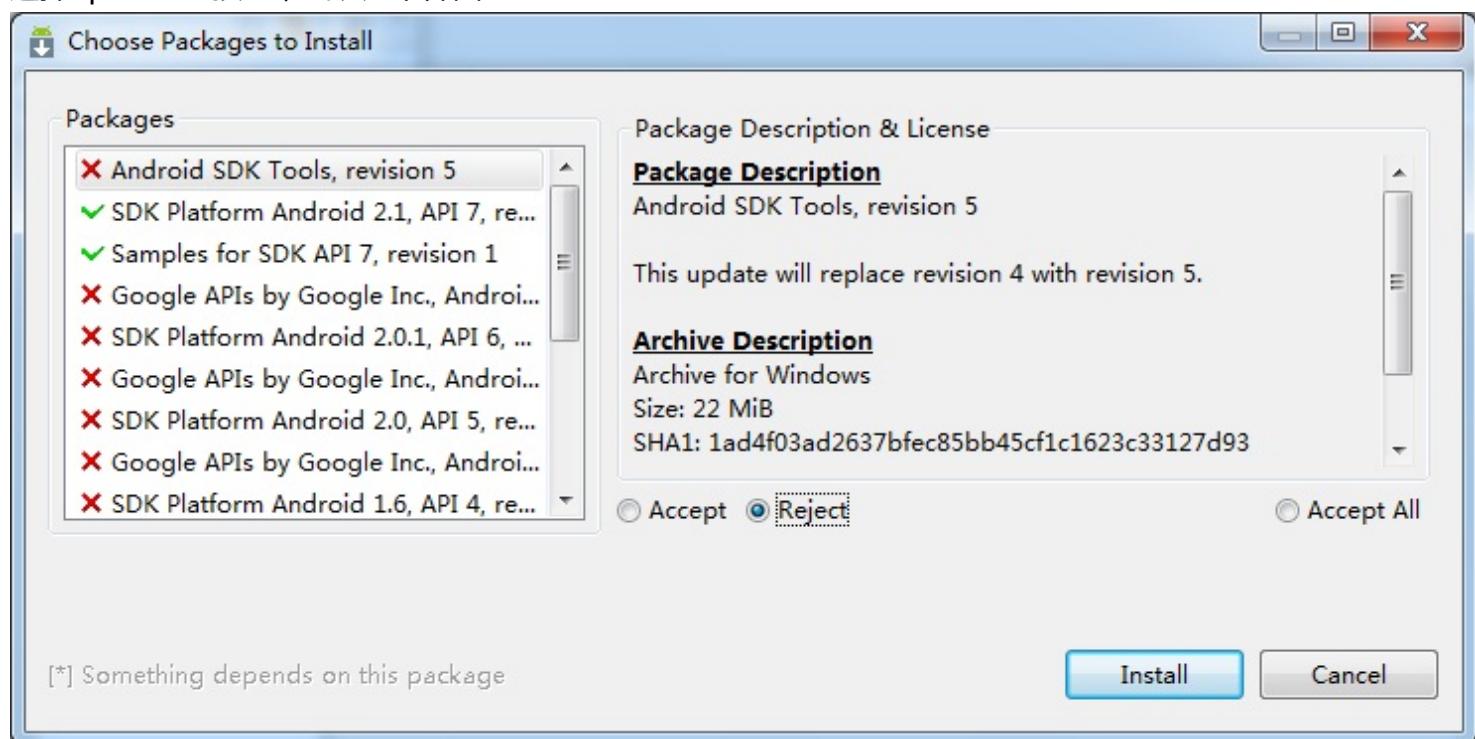


选择你的android SDK解压后的目录，选错了就会报错，这个是升级工具，目前还没有一个版本的SDK

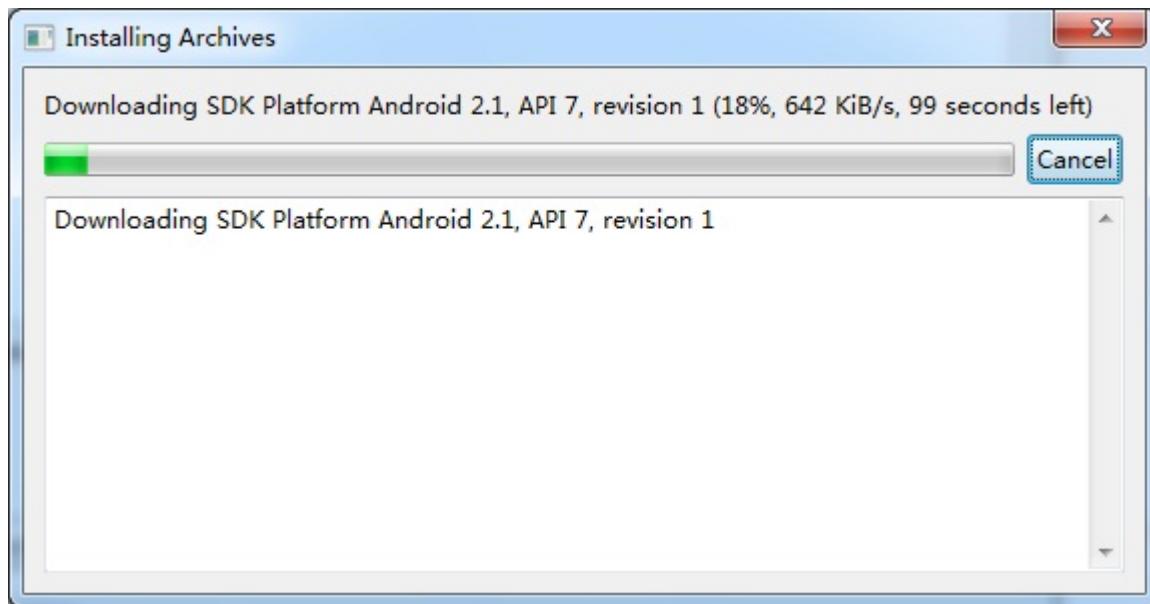
(2) 升级SDK版本,选择菜单 window->Android sdk and avd manager 出现如下界面



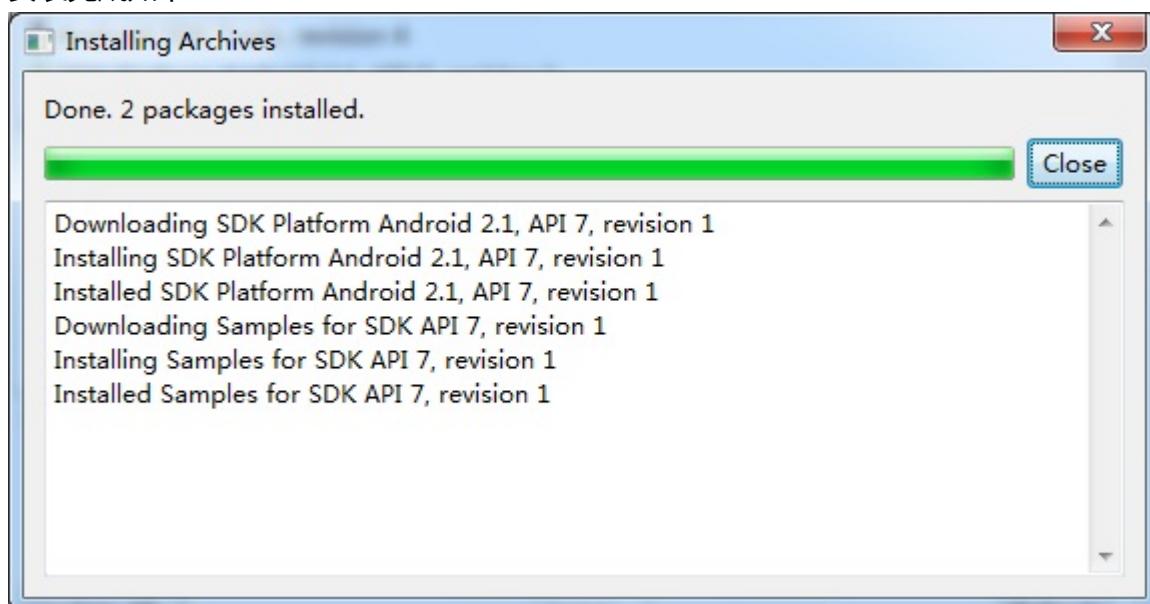
选择update all按钮，出现如下界面



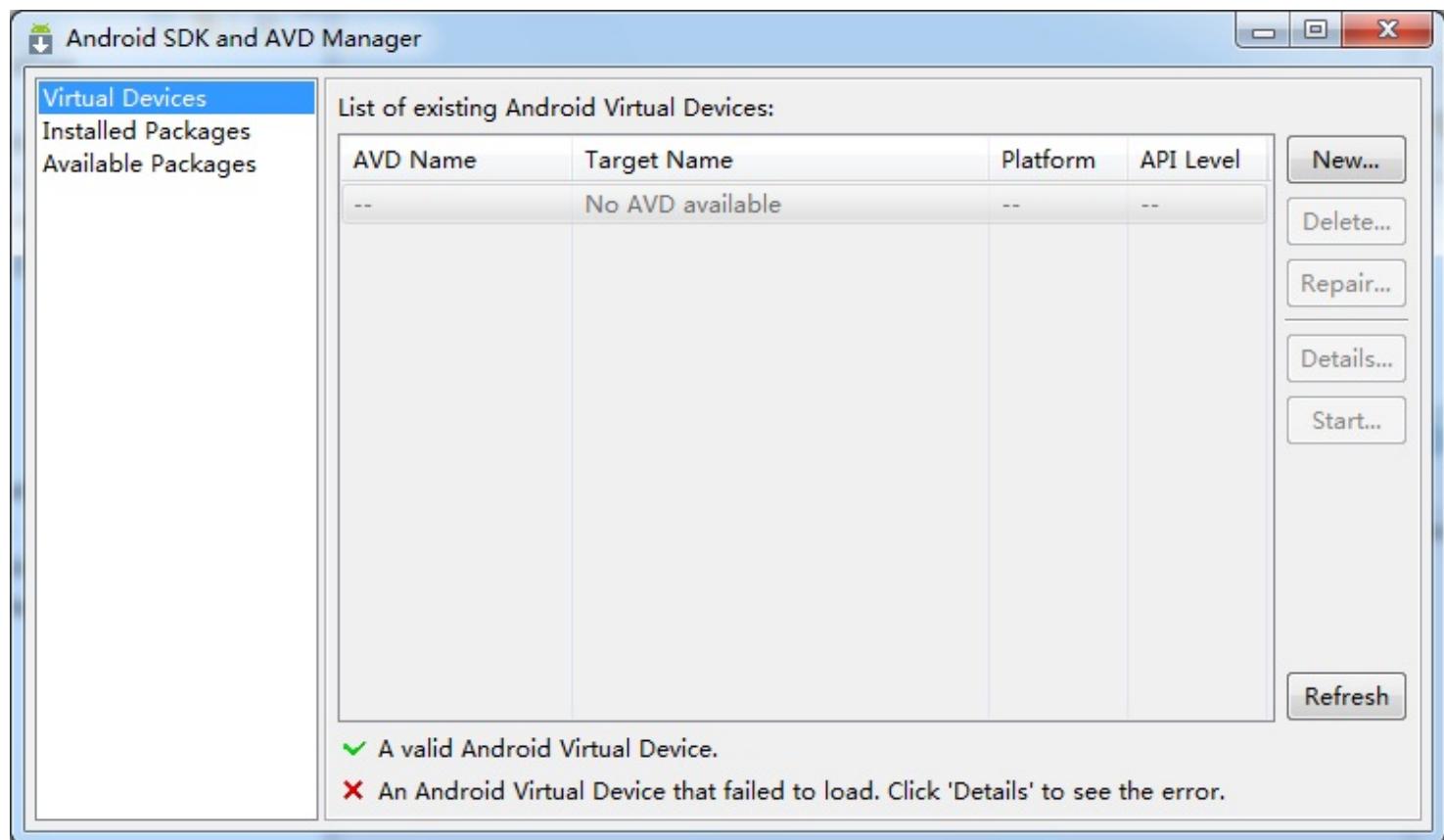
选择左边的某一项，点击accept表示安装，点击reject表示不安装，我这里只选了SDK 2.1 和samples for api 7，自己可以任意自定义，确定后，选择install按钮，进入安装界面如下：



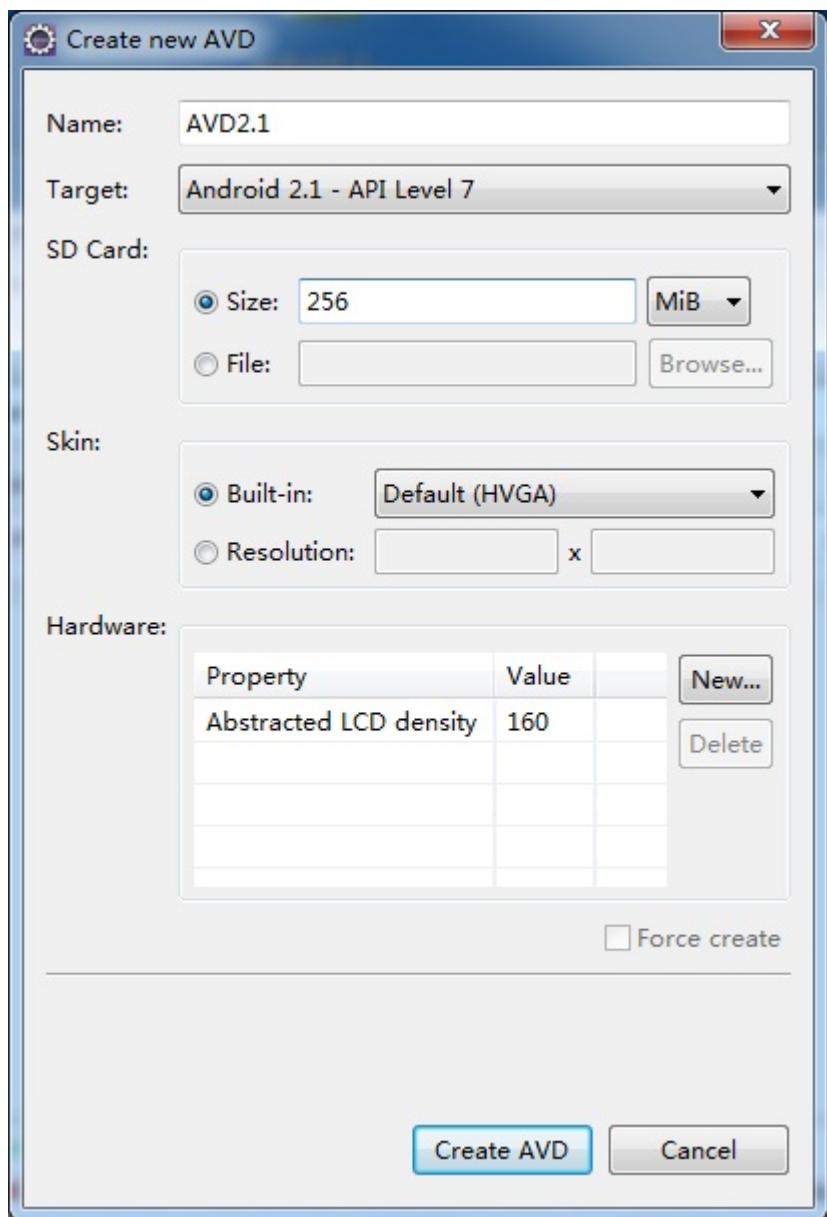
安装完成如下：



(3)新建AVD(android virtual device)和上面一样，进入android sdk and avd manager,选中Virtual Devices在点击New按钮



点击New按钮后，进入如下界面：



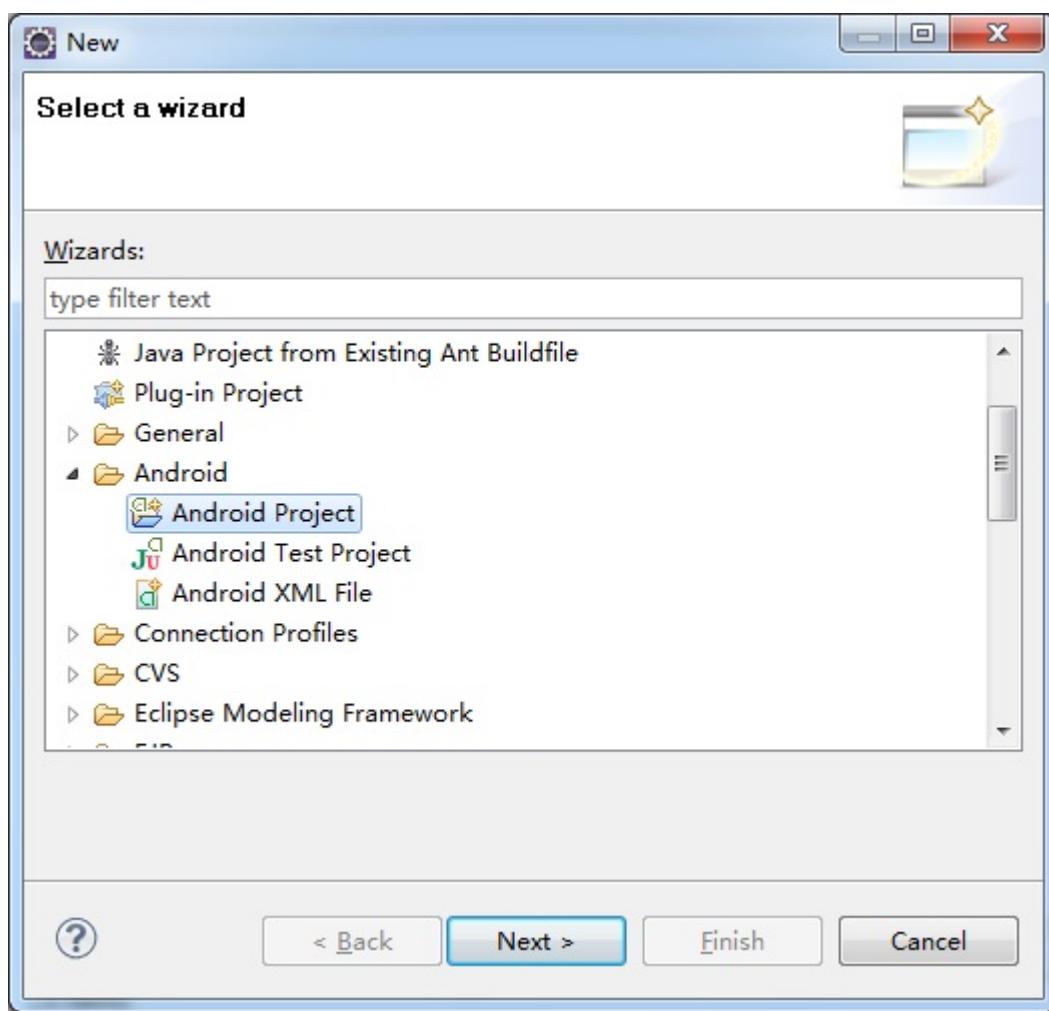
名称可以随便取，target选择你需要的SDK版本，SD卡大小自定义,点击 Create AVD,得到如下结果



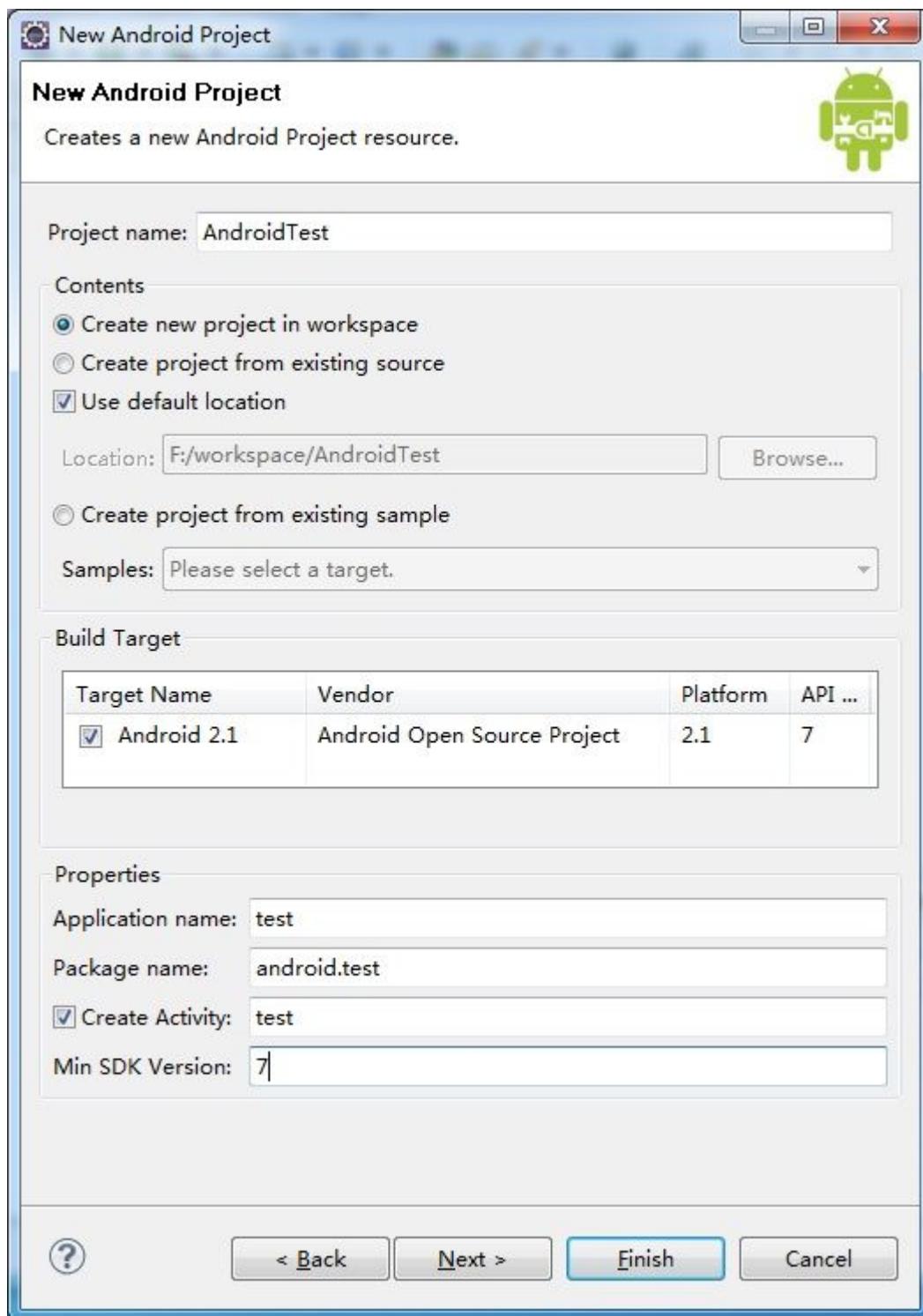
如上显示创建AVD完毕

3 新建Android项目

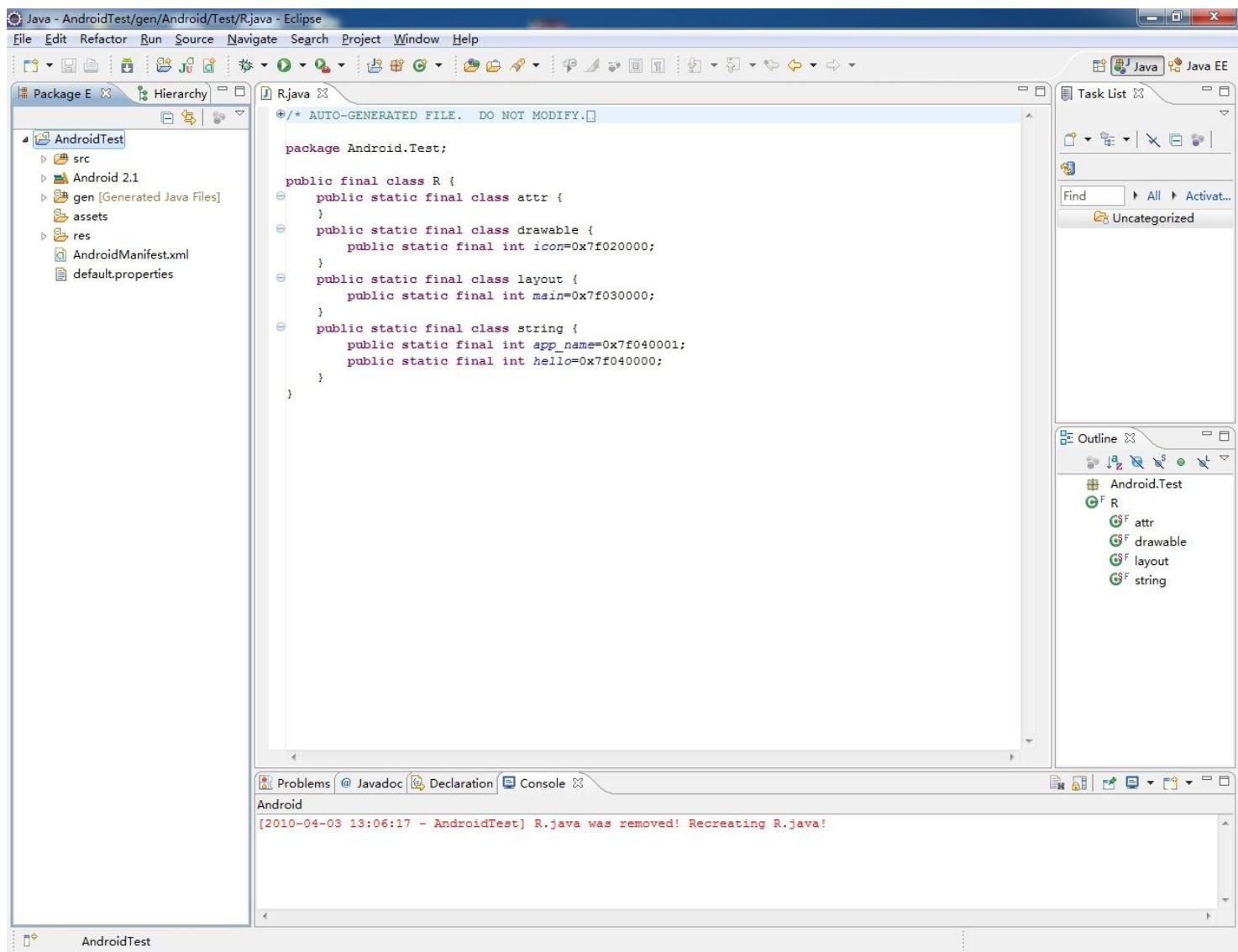
(1)选择菜单file->new->other 进入如下界面:



选择新建Android Project项目，点击Next按钮，进入如下界面



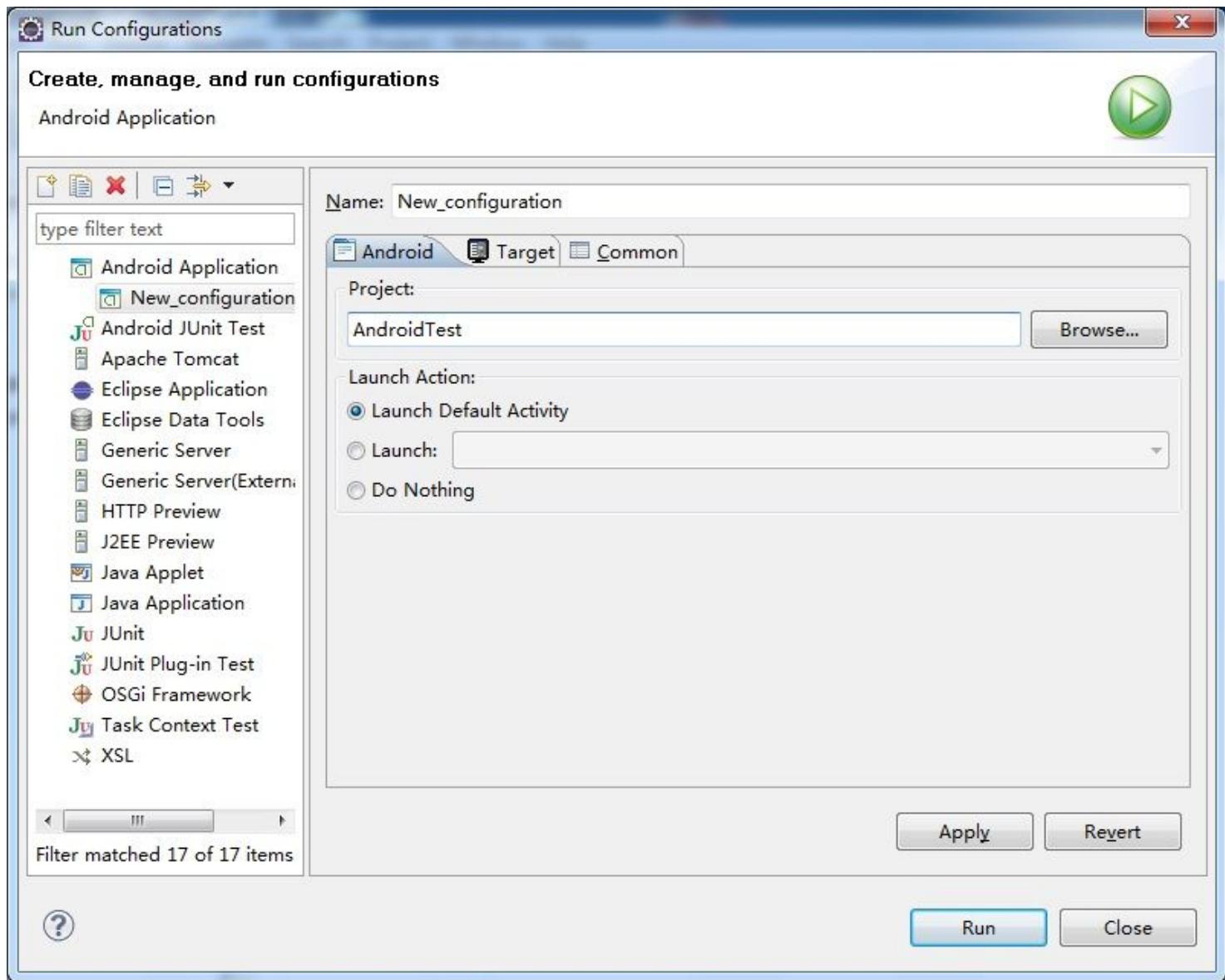
名称自定义，应用程序名自定义，报名必须包含一个点以上，min SDK version里面必须输入整数
点击Next出现如下界面：



注:若有错误如: Project ... is missing required source folder: 'gen' ,则将gen->Android.Test->R.java这个文件删掉 , Eclipse会为我们重新生成这个文件 , 并且不会报错。

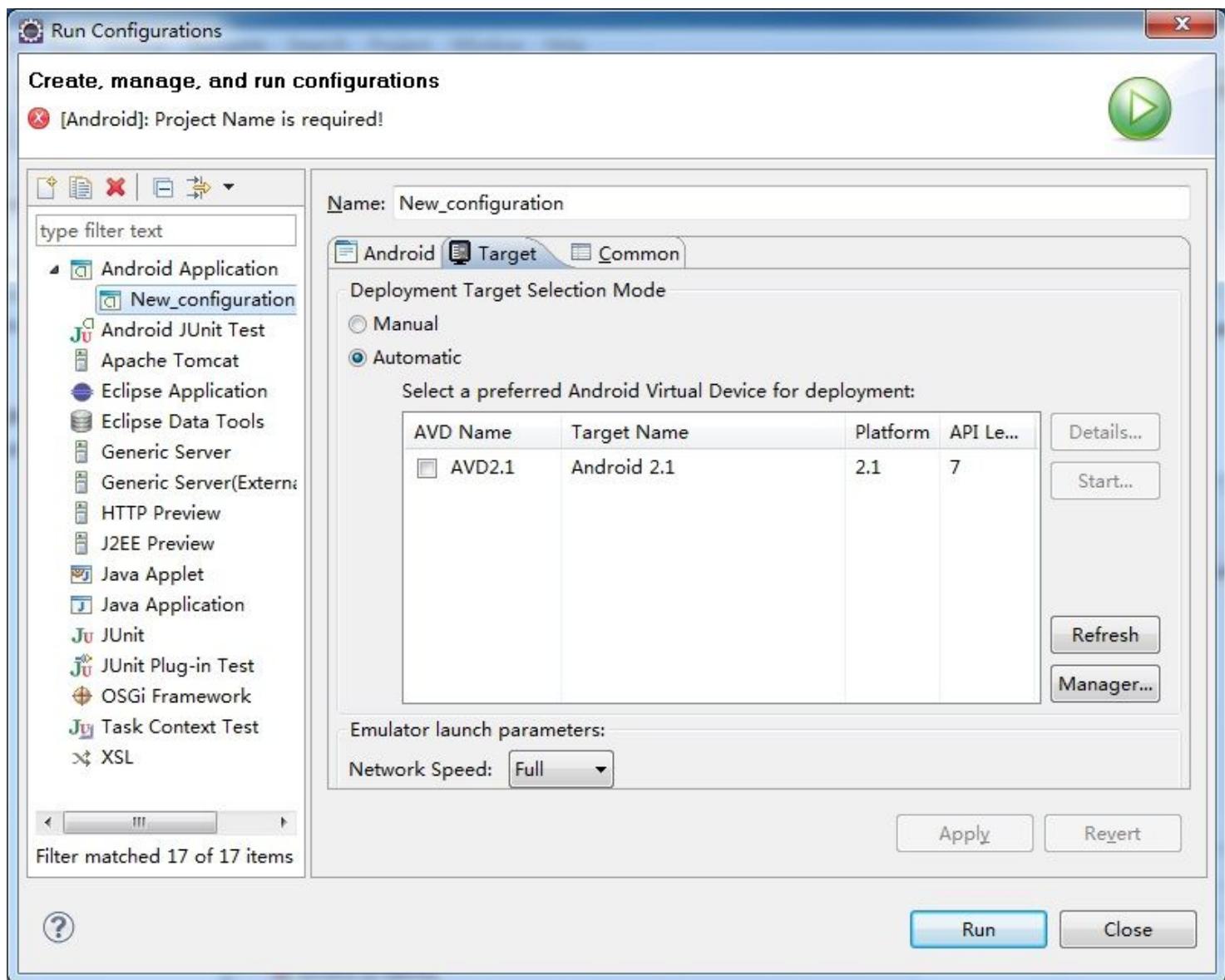
(3)配置运行

右键项目->Run as -> Run Configuration 进入如下界面:



该界面，点击Browse 按钮，选择你要运行的项目

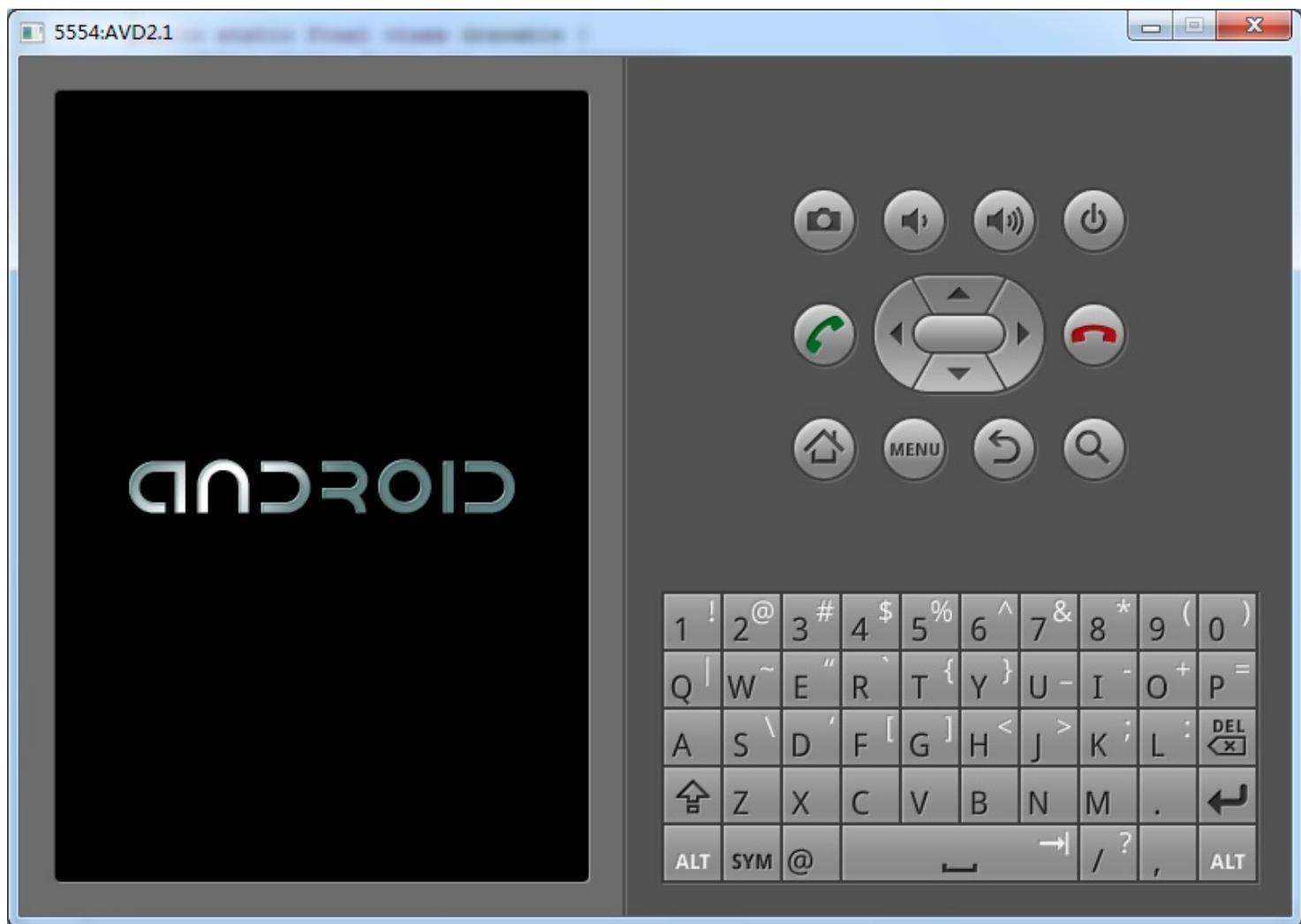
选择Target切换到以下界面



该界面选择运行的AVD，将AVD前面的方框设置为选择状态。

(4) 测试项目运行

右键项目名称->run as ->Android Application 即可启动运行该Android程序，如下所示：



正在进入



测试程序运行结果

四 结束语

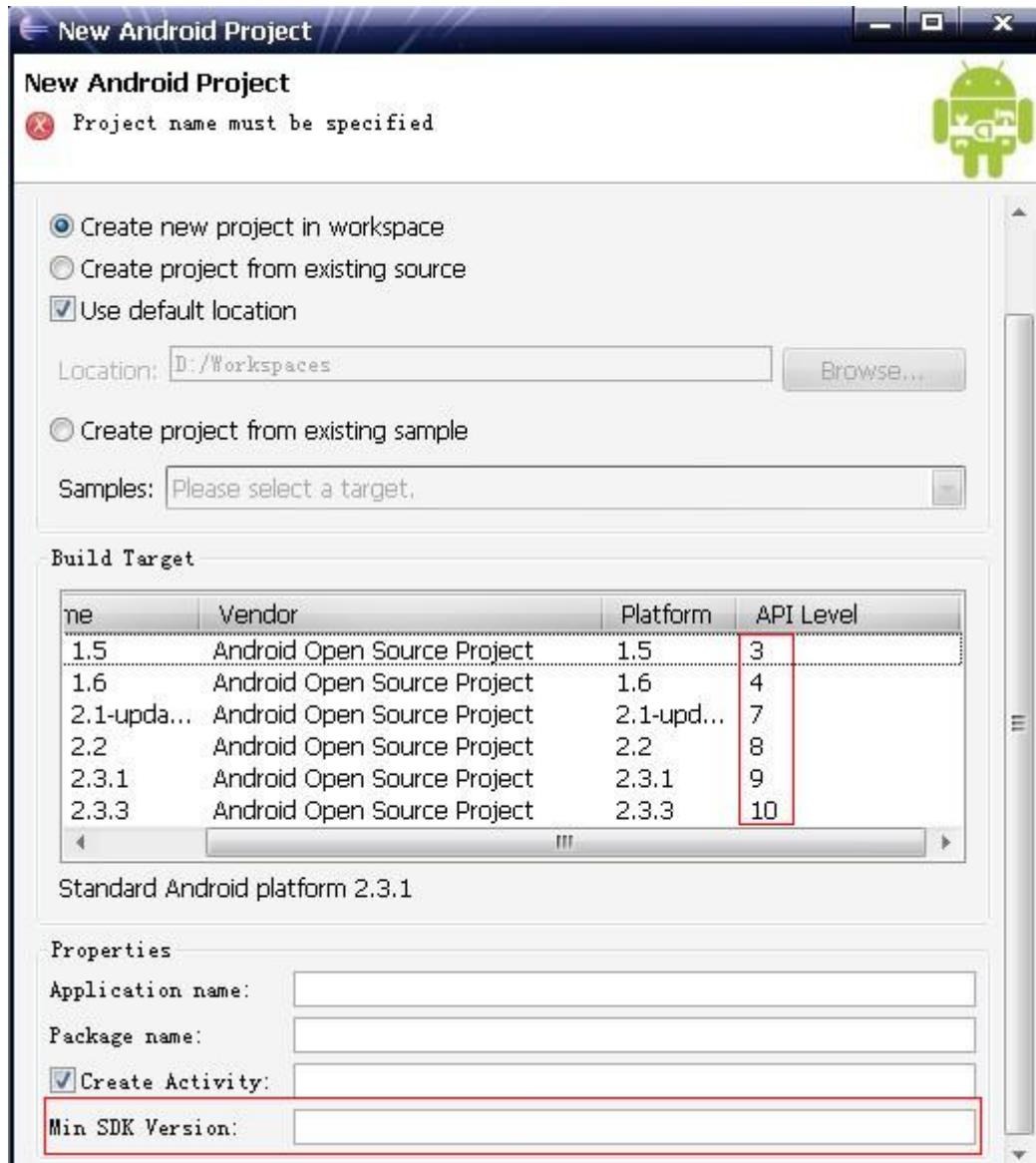
至此，android开发环境搭建完毕，有问题请留言。在这里要注意，我这里只是下载了android sdk r4升级工具，没有下载具体的SDK，而是通过在Eclipse里面的Android Sdk管理工具升级的，你也可以直接下载具体的SDK版本，如: Android sdk 2.1 r1 上面有这个的下载链接，但我任务用升级工具更好。

1.52 Android开发配置需要注意的地方

发表时间: 2011-07-11

Android开发配置需要注意的地方

Min SDK Version(意思是最低限度的系统版本), 一般情况跟对应版本的API版本号一样
如下



AndroidManifest.xml中activity name有个点，意思是说表示这个activity的路径是在manifest标签中声明的package下。这样系统会自动到这个包下去找这个activity。当然你也可以写包名+activity name

```
<activity android:name=".IaiaiActivity"></activity>
```

@符号和?符号来应用资源。

@符号表明了我们应用的资源是前边定义过的(或者在前一个项目中或者在Android 框架中)。问号?表明了我们引用的资源的值在当前的主题当中定义过。通过引用在<item>里边定义的名字可以做到(panelTextColor 用的颜色和panelForegroundColor中定义的一样)。这中技巧只能用在XML资源当中。

```
android:background="?bottom_bg"
```

Activity横屏时ListView会刷新发生变化，如果不想让它刷新变化则在AndroidManifest.xml文件当中的activity后加参数:

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

1.53 Android画图之抗锯齿

发表时间: 2011-07-12

此转

在画图的时候，图片如果旋转或缩放之后，总是会出现那些华丽的锯齿。其实Android自带了解决方式。

方法一：给Paint加上抗锯齿标志。然后将Paint对象作为参数传给canvas的绘制方法。

```
paint.setAntiAlias(true);
```

方法二：给Canvas加上抗锯齿标志。

有些地方不能用paint的，就直接给canvas加抗锯齿，更方便。

```
canvas.setDrawFilter(new PaintFlagsDrawFilter(0, Paint.ANTI_ALIAS_FLAG|Paint.FILTER_BITMAP_FLAG))
```

测试代码如下：

```
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Matrix;
import android.graphics.Paint;
import android.graphics.PaintFlagsDrawFilter;
import android.view.View;

public class MyView extends View {
    private PaintFlagsDrawFilter pfd;
    private Paint mPaint = new Paint();
    private Matrix matrix = new Matrix();
    private Bitmap bmp;

    public MyView(Context context) {
        super(context);
        mPaint.setAntiAlias(true);
        mPaint.setFilterBitmap(true);
        mPaint.setDither(true);
        pfd = new PaintFlagsDrawFilter(0, Paint.ANTI_ALIAS_FLAG|Paint.FILTER_BITMAP_FLAG);
        mPaint.setDrawFilter(pfd);
    }

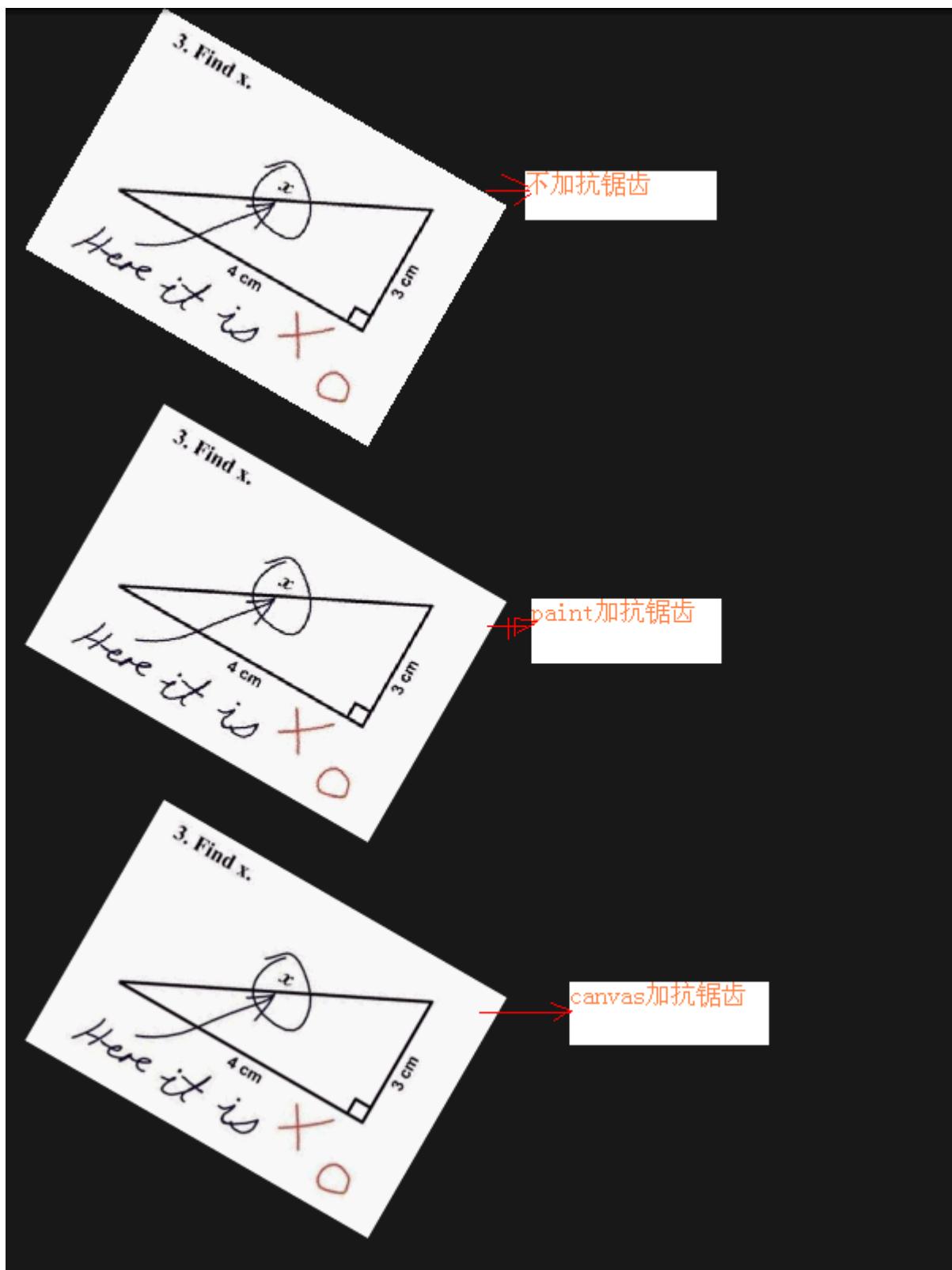
    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        canvas.drawBitmap(bmp, matrix, mPaint);
    }
}
```

```
    initialize();
}

private void initialize() {
    pfd = new PaintFlagsDrawFilter(0, Paint.ANTI_ALIAS_FLAG|Paint.FILTER_BITMAP_FLAG);
    mPaint.setAntiAlias(true);
    matrix.setRotate(30);
    matrix.postScale(0.5f, 0.5f);
    bmp = BitmapFactory.decodeResource(getResources(), R.drawable.show);
}

@Override
public void dispatchDraw(Canvas canvas) {
    canvas.translate(100, 0);
    canvas.drawBitmap(bmp, matrix, null);
    canvas.translate(0, 250);
    canvas.drawBitmap(bmp, matrix, mPaint);
    canvas.setDrawFilter(pfd);
    canvas.translate(0, 250);
    canvas.drawBitmap(bmp, matrix, null);
}
}
```

下图是效果：



可以看出，两种方式都挺有效的。

1.54 Android 组件 SeekBar,RatingBar,Chronometer

发表时间: 2011-07-12

今天补充三个组件的使用，避免日后忘记。它们分别是

SeekBar：用户调整进度的指示进度条

RatingBar：用于显示和调整评分

Chronometer：用户显示时间推移

1、SeekBar

例如我们用播放器看电影的时候，经常会向前移动进度，SeekBar就是这个功能，它类似一个进度条，但是调节器，可以被用户移动。

例如：

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<SeekBar
    android:id="@+id/seekbar"
    android:layout_width="240px"
    android:layout_height="wrap_content"
    android:max="500"
    />
<TextView
    android:id="@+id/text"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    />
</LinearLayout>
```

Test01.java

```
package org.hualang.test01;

import android.app.Activity;
import android.os.Bundle;
import android.widget.SeekBar;
import android.widget.TextView;

public class Test01 extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        SeekBar seek = (SeekBar) findViewById(R.id.seekbar);
        seek.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {

            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {
                // TODO Auto-generated method stub
            }

            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {
                // TODO Auto-generated method stub
            }

            @Override
            public void onProgressChanged(SeekBar seekBar, int progress,
                boolean fromUser) {
                // TODO Auto-generated method stub
                ((TextView) findViewById(R.id.text)).setText("Value:" + progress);
                seekBar.setSecondaryProgress((progress + seekBar.getMax()) / 2);
            }
        });
    }
}
```

```
    }  
}
```

运行结果如下：



2、RatingBar

这个组件主要是用于显示或从用户那里进行数值设定，默认的情况下，进度条里有5个五角星，而且它类似SeekBar，也可以进行拖动进度条

需要注意的是，不同于SeekBar，实现的onRatingChange()方法是在修改完成之后调用的，通常是在用户抬起手指时候，也就是说，当用户在五角星上拖动设置评分的时候，这个方法并不会被调用，而是在用户停止触摸控件的时候获得调用。

main.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    >  
<RatingBar  
    android:id="@+id/ratebar1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:numStars="5"  
    android:stepSize="0.25"  
    />  
<TextView
```

```
    android:id="@+id/text"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"

/>
</LinearLayout>
```

MainActivity.java

```
package org.hualang.rating;

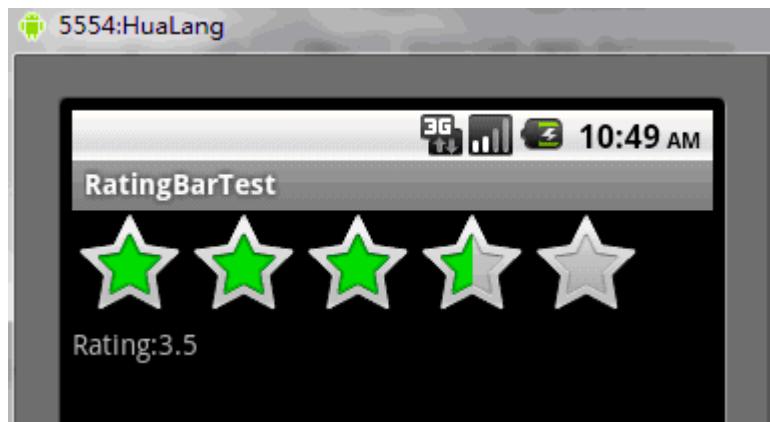
import android.app.Activity;
import android.os.Bundle;
import android.widget.RatingBar;
import android.widget.TextView;

public class MainActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        RatingBar rate = (RatingBar) findViewById(R.id.ratebar1);
        final TextView text = (TextView) findViewById(R.id.text);
        rate.setOnRatingBarChangeListener(new RatingBar.OnRatingBarChangeListener() {

            @Override
            public void onRatingChanged(RatingBar ratingBar, float rating,
                boolean fromUser) {
                text.setText("Rating:"+rating);
            }
        });
    }
}
```

```
    }  
}
```

运行结果：



3、Chronometer

当你想要显示时间的推移而不是进程的增加的时候，可以使用Chronometer作为一个计时器。当需要记录用户进行某项操作所消耗的时间的时候，或者在游戏中需要限制时间时候，它将非常实用。

这里，Chronometer对象的format属性可以用来设置显示时间周围的文字。并且只有它的start()在被调用之后才会显示时间及附加文字，要停止计时器可以调用stop()方法。setBase()方法用来设定设置计时器的起点，当将计时器几点设置为0的时候，计时器将从手机上次重启的时刻开始计时。

main.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    >  
<Chronometer  
    android:id="@+id/ChronometerTest"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:format="Timer:%s"  
/>>
```

```
<TextView
    android:id="@+id/text01"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
/>
</LinearLayout>
```

MainActivity.java

```
package org.hualang.chronometer;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.widget.Chronometer;
import android.widget.TextView;

public class MainActivity extends Activity {
    /** Called when the activity is first created. */
    private Chronometer timer;
    private TextView text;
    String DEBUG_TAG;

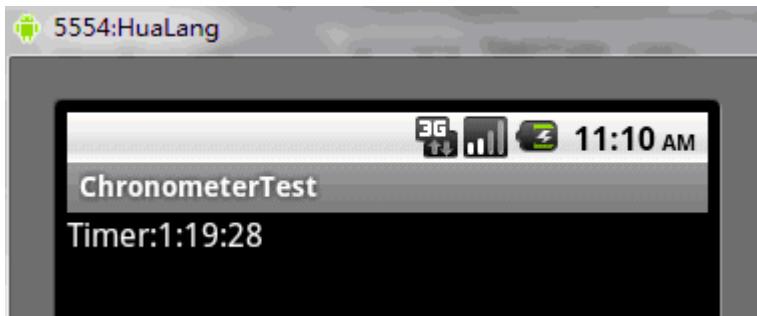
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        text = (TextView) findViewById(R.id.text01);
        timer = (Chronometer) findViewById(R.id.ChronometerTest);
        long base = timer.getBase();
        Log.d(DEBUG_TAG, "BASE="+base);
        timer.setBase(0);
        timer.start();

        timer.setOnChronometerTickListener(new Chronometer.OnChronometerTickListener() {
```

```
    @Override
    public void onChronometerTick(Chronometer chronometer) {
        // TODO Auto-generated method stub
    }
});
```

}

运行结果：



1.55 Android之SharedPreferences

发表时间: 2011-07-13

SharedPreferences存储类效率分析

SharedPreferences是Android平台上一个轻量级的存储类，主要是保存一些常用的配置比如窗口状态，一般在Activity中重载窗口状态onSaveInstanceState保存一般使用SharedPreferences完成，它提供了Android平台常规的Long长整形、Int整形、String字符串型的保存，它是什么样的处理方式呢？

SharedPreferences类似过去Windows系统上的ini配置文件，但是它分为多种权限，可以全局共享访问，android123提示最终是以xml方式来保存，整体效率来看不是特别的高，对于常规的轻量级而言比SQLite要好不少，如果真的存储量不大可以考虑自己定义文件格式。xml 处理时Dalvik会通过自带底层的本地XML Parser解析，比如XMLpull方式，这样对于内存资源占用比较好。

Android数据存取之Preferences

这种方式应该是用起来最简单的Android读写外部数据的方法了。他的用法基本上和J2SE(java.util.prefs.Preferences)中的用法一样，以一种简单、透明的方式来保存一些用户个性化设置的字体、颜色、位置等参数信息。一般的应用程序都会提供“设置”或者“首选项”的这样的界面，那么这些设置最后就可以通过Preferences来保存，而程序员不需要知道它到底以什么形式保存的，保存在了什么地方。当然，如果你愿意保存其他的东西，也没有什么限制。只是在性能上不知道会有什么问题。

在Android系统中，这些信息以XML文件的形式保存在 /data/data/PACKAGE_NAME/shared_prefs 目录下。

数据读取：

```
String PREFS_NAME = "com.iaiai";
SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);
boolean iaiai = settings.getBoolean("iaiai", false);
String hello = settings.getString("hello", "Hi");
```

这段代码中：

```
SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);
```

通过名称，得到一个SharedPreferences，顾名思义，这个Preferences是共享的，共享的范围据现在同一个Package中，这里面说所的Package和Java里面的那个Package不同，貌似这里面的Package是指在AndroidManifest.xml文件中：

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.iaiai"
  android:versionCode="1"
  android:versionName="1.0.0">
```

这里面的package。根据我目前的实验结果看，是这样的，欢迎指正。后面的那个int是用来声明读写模式，先不管那么多了，暂时就知道设为0(android.content.Context.MODE_PRIVATE)就可以了。

```
boolean silent = settings.getBoolean("iaiai", false);
```

获得一个boolean值，这里就会看到用Preferences的好处了：可以提供一个缺省值。也就是说如果Preference中不存在这个值的话，那么就用后面的值作为返回值，这样就省去了我们的if什么什么为空的判断。

数据写入：

```
String PREFS_NAME = "com.iaiai";
SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);
SharedPreferences.Editor editor = settings.edit();
editor.putBoolean("iaiai", true);
editor.putString("hello", "Hello~");
editor.commit();
```

有了上面数据读取的代码，这里面的就容易理解了，只是别忘了最后的commit()；

```
//得到SharedPreferences中的值
SharedPreferences passwdfile = getSharedPreferences("com.iaiai", 0);
String im = passwdfile.getString("im", null);
TextView tv = (TextView) findViewById(R.id.tv);
```

```
tv.setText("choice: " + im);

//编辑SharePreferences中的值
Editor passwdfile = getSharedPreferences("com.iaiai", 0).edit();
passwdfile.putString("im",item.getTitle().toString());
passwdfile.commit();
finish();
```

1.56 Android之TableLayout

发表时间: 2011-07-13

TableLayout和我们平时在网页上见到的Table有所不同，TableLayout没有边框的，它是由多个TableRow对象组成，每个TableRow可以有0个或多个单元格，每个单元格就是一个View。这些TableRow，单元格不能设置layout_width,宽度默认是fill_parent的，只有高度layout_height可以自定义，默认是wrap_content。

单元格可以为empty，并且通过android:layout_column可以设置index值实现跳开某些单元格。在TableRow之间，添加View,设置layout_height以及背景色，就可以实现一条间隔线。android:layout_span可以设置合并几个单元格：

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    android:stretchColumns="1">
    <TableRow>
        <TextView android:text="用户名" android:textStyle="bold"
            android:gravity="right" android:padding="3dip" />
        <EditText android:id="@+id/username" android:text="用户名"
            android:padding="3dip" android:scrollHorizontally="true" />
    </TableRow>
    <TableRow>
        <TextView android:text="密码" android:textStyle="bold"
            android:gravity="right" android:padding="3dip" />
        <EditText android:id="@+id/password" android:text="密码"
            android:password="true" android:padding="3dip"
            android:scrollHorizontally="true" />
    </TableRow>
    <TableRow android:gravity="right">
        <Button android:id="@+id/cancel" android:text="cancel" />
        <Button android:id="@+id/login" android:text="login" />
    </TableRow>
    <!-- 间隔线 -->
    <View android:layout_height="2dip" android:background="#F00" />
    <View android:layout_height="5dip" />
    <TableRow>
        <TextView android:text="合并2个单元格" android:layout_span="2"
```

```
        android:gravity="center_horizontal" android:background="#FFC0C0C0"
        android:textColor="#f00" android:padding="3dip" />
    </TableRow>
</TableLayout>
```

运行效果:

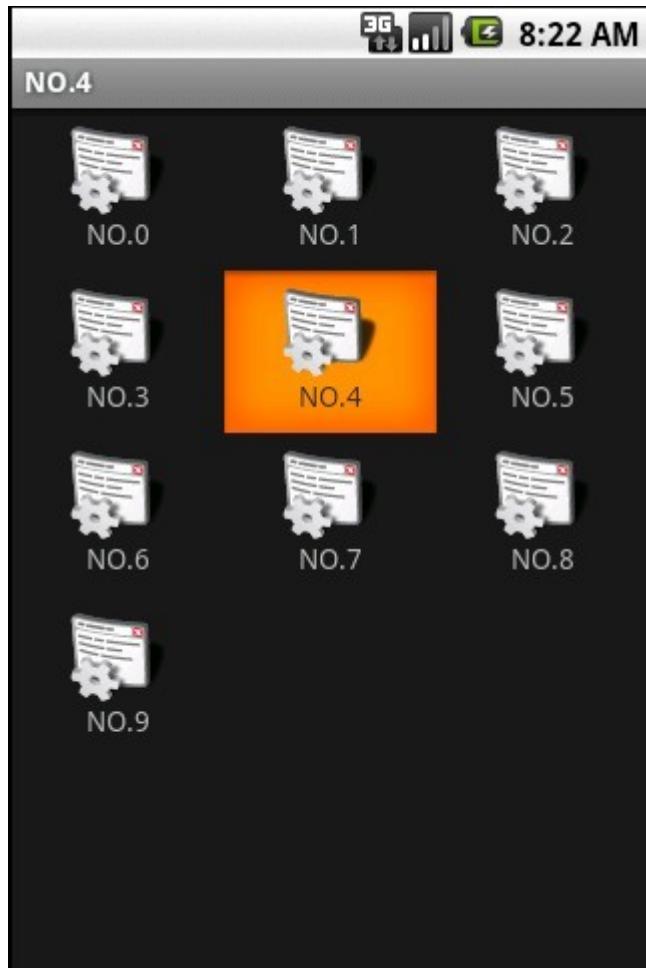


1.57 Android之GridView(九宫图)

发表时间: 2011-07-13

GridView跟ListView都是比较常用的多控件布局，而GridView更是实现九宫图的首选!本文就是介绍如何使用GridView实现九宫图。GridView的用法很多，网上介绍最多的方法就是自己实现一个ImageAdapter继承BaseAdapter，再供GridView使用，类似这种方法本文不再重复。

先来贴出本文代码运行的结果：



main.xml源代码如下，本身是个GridView，用于装载Item：

```
<?xml version="1.0" encoding="utf-8"?>
<GridView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/gridview" android:layout_width="fill_parent"
    android:layout_height="fill_parent" android:numColumns="auto_fit"
    android:verticalSpacing="10dp" android:horizontalSpacing="10dp"
    android:columnWidth="90dp" android:stretchMode="columnWidth"
    android:gravity="center" />
```

介绍一下里面的某些属性：

android:numColumns="auto_fit" , GridView的列数设置为自动
android:columnWidth="90dp" , 每列的宽度，也就是Item的宽度
android:stretchMode="columnWidth" , 缩放与列宽大小同步
android:verticalSpacing="10dp" , 两行之间的边距，如：行一(NO.0~NO.2)与行二(NO.3~NO.5)间距为10dp
android:horizontalSpacing="10dp" , 两列之间的边距。

gridview.xml布局：

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="wrap_content" android:paddingBottom="4dip"
    android:layout_width="fill_parent">
    <ImageView android:layout_height="wrap_content" android:id="@+id/ItemImage"
        android:layout_width="wrap_content" android:layout_centerHorizontal="true">
    </ImageView>
    <TextView android:layout_width="wrap_content"
        android:layout_below="@+id/ItemImage" android:layout_height="wrap_content"
        android:text="TextView01" android:layout_centerHorizontal="true"
        android:id="@+id/ItemText">
    </TextView>
</RelativeLayout>
```

最后就是JAVA的源代码了，

```
@Override
protected void onHandle(Bundle savedInstanceState) {
    setContentView(R.layout.main);

    // 列表
    GridView gridview = (GridView) findViewById(R.id.gridView);

    // 生成适配器
    SimpleAdapter adapter = getSimpleAdapter();
```

```
// 添加并且显示
gridview.setAdapter(adapter);
// 添加消息处理
gridview.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
                           long arg3) {
        // 在本例中arg2=arg3
        Map<String, Object> item = (Map<String, Object>) arg0
            .getItemAtPosition(arg2);
        // 显示所选Item的ItemText
        setTitle(item.get(Attribute.GRID_VIEW_ID).toString());
    }
});
```

```
}

/**
 * 获取数据生成适配器
 *
 * @return
 */
private SimpleAdapter getSimpleAdapter() {
    // 生成动态数组，并且转入数据
    List<Map<String, Object>> grid = new ArrayList<Map<String, Object>>();
    for (int i = 0; i < 10; i++) {
        Map<String, Object> map = new HashMap<String, Object>();
        map.put(Attribute.GRID_VIEW_ID, i + 1);
        map.put(Attribute.GRID_VIEW_IMG, R.drawable.icon); // 添加图像资源的ID
        map.put(Attribute.GRID_VIEW_TEXT, "NO." + String.valueOf(i)); // 按序号估
        grid.add(map);
    }

    // 生成适配器
    SimpleAdapter adapter = new SimpleAdapter(this, grid,
        R.layout.gridview, new String[] { Attribute.GRID_VIEW_IMG,
                                         Attribute.GRID_VIEW_TEXT }, new int[] { R.id.I
                                         R.id.ItemText });
}
```

```
        return adapter;  
    }
```

1.58 android 屏幕分辨率问题

发表时间: 2011-07-14

iPhone的icon设计就这么简单，iPhone的屏幕只有320×480像素，所以程序没有适配性问题。

Android

在研究 Android的icon设计之前，有必要先了解Android的界面是如何适配多样化屏幕的。

适配性

由于同一个UI元素（如100 x100像素的图片）在高精度的屏幕上要比低精度的屏幕上看起来要小，为了让这两个屏幕上的图片看起来效果差不多，可以采用以下两种方法：

程序将图片进行缩放，但是效果较差。

为这两个精度屏幕的手机各提供一个图片。

但是屏幕的参数多样化，为每一个精度的屏幕都设计一套icon，工作量大并且不能满足程序的兼容性要求，势必要对屏幕的分级，如在160dpi和180dpi的手机屏幕上采用同一套icon，当这套icon在240dpi效果满足不了设计要求，就需要另做一套稍大些的icon。

在 Android 1.5以及更早的版本中，只支持3.2" 屏幕上的HVGA (320×480)分辨率，开发人员也不需要考虑界面的适配性问题。从Android 1.6之后，平台支持多种尺寸和分辨率的设备，这也就意味着开发人员在设计时要考虑到屏幕的多样性。

为了简化设计并且兼容更多的手机屏幕，平台依照尺寸和分辨率对屏幕进行了区分：

三种尺寸：大，中，小。

三种精度：高 (hdpi)，中 (mdpi) 和低 (ldpi)。

程序可以为这三种尺寸的屏幕提供默认资源，如有需要，还可以为各种精度的屏幕提供资源。在运行时，系统会根据屏幕布局加载正确尺寸或者精度的图片。

Icon Type	Standard Asset Sizes (in Pixels), for Generalized Screen Densities		
	Low density screen (<i>ldpi</i>)	Medium density screen (<i>mdpi</i>)	High density screen (<i>hdpi</i>)
Launcher	36 x 36 px	48 x 48 px	72 x 72 px
Menu	36 x 36 px	48 x 48 px	72 x 72 px
Status Bar	24 x 24 px	32 x 32 px	48 x 48 px
Tab	24 x 24 px	32 x 32 px	48 x 48 px
Dialog	24 x 24 px	32 x 32 px	48 x 48 px
List View	24 x 24 px	32 x 32 px	48 x 48 px

Android的icon尺寸

在运行时，程序为最佳显示效果提供了三种方法：

1.图片缩放

基于当前屏幕的 精度，平台自动加载任何未经缩放的限定尺寸和精度的图片。如果图片不匹配，平台会加载默认资源并且在放大或者缩小之后可以满足当前界面的显示要求。例如，当前为高精度屏幕，平台会加载高精度资源（如图片），如果没有，平台会将中精度资源缩放至高精度。

2.自动定义像素尺寸和位置

如果程序不支持多种精度屏幕，平台会自动 定义像素绝对位置和尺寸值等，这样就能保证元素能和精度160的屏幕上一样能显示出同样尺寸的效果。例如，要让WVGA高精度屏幕和传统的HVGA 屏幕一样显示同样尺寸的图片，当程序不支持时，系统会对程序谎称屏幕分辨率为320×533，在 (10 , 10) 到 (100 , 100) 的区域内绘制图形完成之后，系统会将图形放大到 (15 , 15) 到 (150 , 150) 的屏幕显示区域。

3 . 兼容更大尺寸的屏幕

当前屏幕超过程序所支持屏幕的上限时，定义`supports-screens`元素，这样超出显示的基准线时，平台在此显示黑色的背景图。例如，WVGA 中精度屏幕上，如程序不支持这样的大屏幕，系统会谎称是一个320×480的，多余的显示区域会被填充成黑色。

但为了达到最佳的显示效果，最好的方法还是设计多套图片。那就有必要对于所有的屏幕依据精度值进行分级（高中低），之后再设计三套icon。

先为主流的中精度屏幕（HVGA）设计一套icon，确定图片的像素尺寸。

为高精度屏幕将图片放大到150%，为低精度屏幕将图片缩小至75%。

将这三套资源放置到程序的三个文件夹下：res/drawable-mdpi/、res/drawable-hdpi/、res/drawable-ldpi/。程序在运行时，平台会根据屏幕的精度调取合适的icon。

设计指南



Android标准icon

Launcher Icon是程序主界面上的功能图标，Android对于各个icon的风格有一定的要求：

符合当下的流行趋势，避免过度使用隐喻。

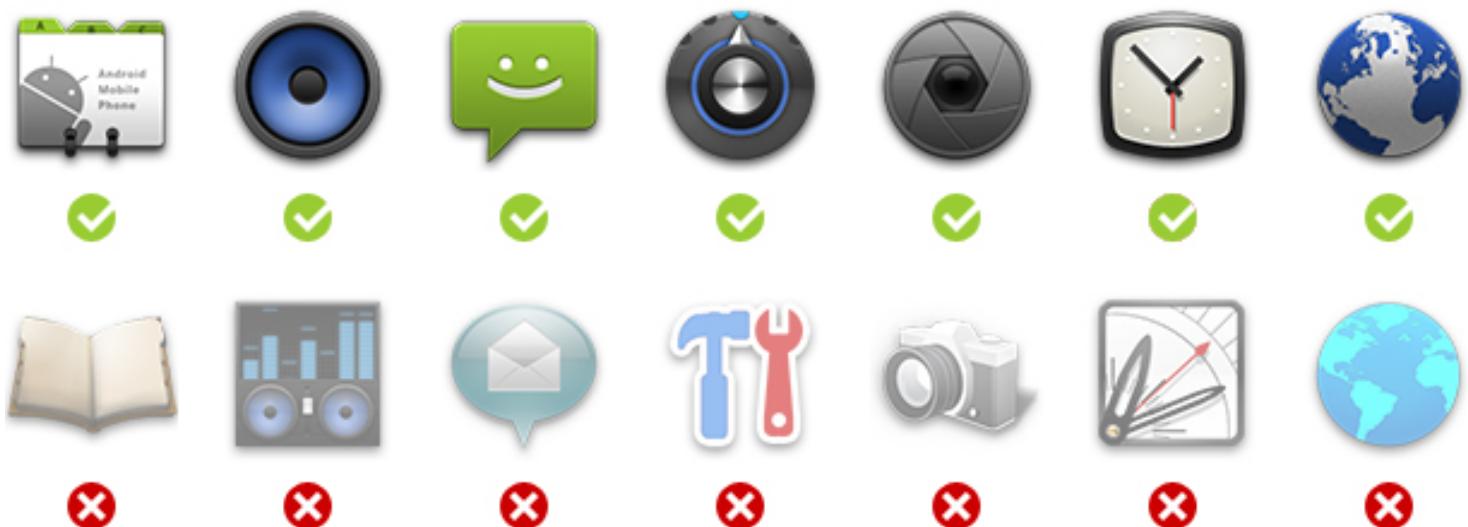
高度简化和夸张，小尺寸图标也能易于识别，不宜太复杂。

尝试抓住程序的主要特征，比如音像作为音乐的icon。

使用自然的轮廓和形状，看起来几何化和有机化，不失真实感。

Icon采用前视角，几乎没有透视，光源在顶部。

不光滑但富有质感。



正确和错误的icon

另外，所有的icon都有文字标签，不要在设计时把文字也放到icon中。

Launcher Icon通常是一个较大的标准图标中包含一个小图标，使用一个中性色彩和一个主要色彩，并保持高度的对比，不宜过度饱和。



推荐色值

尺寸和定位

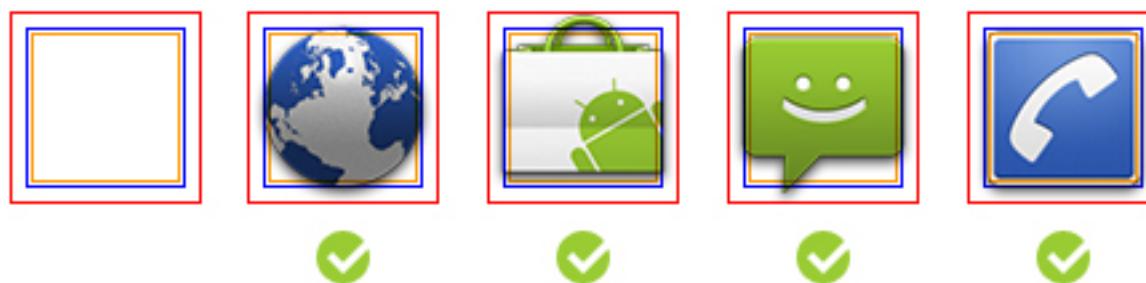
Launcher icons要有多样化的形状和样式，但又要形成统一的视觉风格，其尺寸和定位也用统一：

(Full Asset) 红色边框为图标尺寸

(Icon) 蓝色边框是图形尺寸，比图标尺寸稍小，图形之外的空间用于显示阴影和特殊效果。

(Square Icon) 橙色边框是另外一种图形尺寸。两种类型的图形尺寸可以达到统一的视觉权重。

Icon dimensions for high-density (hdpi) screens:

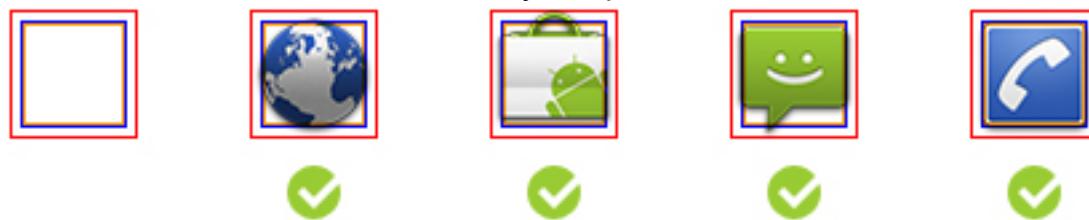


Full Asset: 72 x 72 px

Icon: 60 x 60 px

Square Icon: 56 x 56 px

Icon Dimensions for medium-density (mdpi) screens:



Full Asset: 48 x 48 px

Icon: 40 x 40 px

Square Icon: 38 x 38 px

Icon Dimensions for low-density (ldpi) screens:



Full Asset: 36 x 36 px

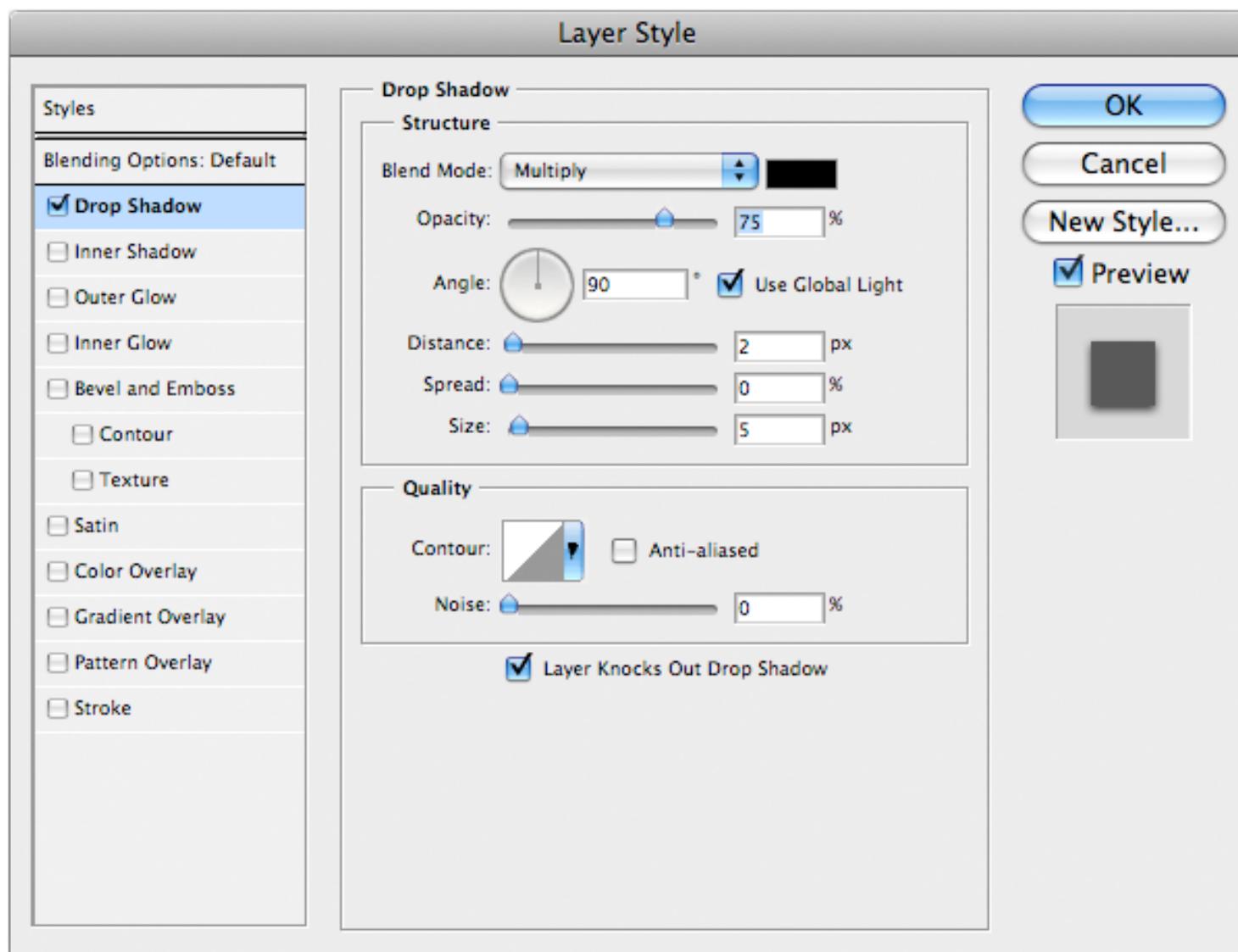
Icon: 30 x 30 px

Square Icon: 28 x 28 px

模板

下载Android的标准图标模板，在此基础上绘制icon，后期还要加上阴影效果。

WVGA (高精度) 屏幕的阴影:



Effect: Drop Shadow

Color: #000000

Blend Mode: Multiply

Opacity: 75%

Angle: 90°

Distance: 2px

Spread: 0%

Size: 5px

另外，还有Menu icon、Status bar icon、Tab icon、Dialog icon 和List view icon的设计教程请查看原文吧。

非常详尽的icon设计教程，一个程序里要放置三套icon，还有Android硬件比较好，不然 内存爆掉

1.59 Android中Activity共享变量的另一方法：Application context

发表时间: 2011-07-14

感觉用这种方式存储登录信息比较好，比SharedPreferences存储登录信息的状态比较合适。

相当于存储session的数据(打个比喻)。

多个Activity中经常使用同一变量时，使用Bundle则比较麻烦，每次调用Activity都需要设置一次。如想在整个应用中使用，在java中一般是使用静态变量，而在android中有个更优雅的方式是使用Application context。用SharedPreferences存储的话关闭应用的时候这个值还是最后一次设置的值，如果是非正常退出程序时登录状态就会有问题，所以在使用上感觉application context是最合适存储当前登录状态的信息，当然可是存储一些相当于session当中存储的值，因为关闭应用后这些值就没有了。

新建一个类，继承自Application

```
package org.mdx.core.modules;

import android.app.Application;

/**
 *
 * <p>
 * Title: ApplicationHelper.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-7-14
 * </p>
 *
 * @author 丸子
```

```
* @version 0.0.1
*/
public class ApplicationHelper extends Application {

    private boolean session_enter = false; // 默认为false

    /**
     * 设置session状态
     *
     * @param bol
     *         true为已登录，false未登录
     */
    public void putSession(boolean bol) {
        session_enter = bol;
    }

    /**
     * 是否登录
     *
     * @return true是，false否
     */
    public boolean isEnter() {
        return session_enter;
    }
}
```

在AndroidManifest.xml的application加个name属性就可以了，如下面所示：

```
<application android:name=".core.modules.ApplicationHelper"
            android:icon="@drawable/icon" android:label="@string/app_name">
```

activity使用时：（这个是重点）

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    getApplicationContext().putSession(false); // 设置登录状态为退出
}
public ApplicationHelper getApplicationContext() {
    return ((ApplicationHelper) super.getApplicationContext());
}
```

1.60 Android布局文件中的include

发表时间: 2011-07-14

居然没有发现，在xml文件中可以使用include，我晕
一下为son.xml文件

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <Button
        android:text="@+id/Button01"
        android:id="@+id/Button01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </Button>
    <Button
        android:text="@+id/Button01"
        android:id="@+id/Button02"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </Button>
</LinearLayout>
```

接下来在main.xml文件中将其include进来，如下

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

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
  />
  <include layout="@layout/son"/>
</LinearLayout>
```

运行效果不解释了。本来还在想，如果include2次，那么如何区分包含进来的控件的id呢，后来试了一下发现，不管include几次，其实都只相当于include了一次。

1.61 Android屏幕分辨率详解 (VGA、HVGA、QVGA、WVGA、WQVGA)

发表时间: 2011-07-15

这些术语都是指屏幕的分辨率。

VGA : Video Graphics Array , 即 : 显示绘图矩阵 , 相当于 640×480 像素 ;

HVGA : Half-size VGA ; 即 : VGA 的一半 , 分辨率为 480×320 ;

QVGA : Quarter VGA ; 即 : VGA 的四分之一 , 分辨率为 320×240 ;

WVGA : Wide Video Graphics Array ; 即 : 扩大的 VGA , 分辨率为 800×480 像素 ;

WQVGA : Wide Quarter VGA ; 即 : 扩大的 QVGA , 分辨率比 QVGA 高 , 比 VGA 低 , 一般是 : 400×240 , 480×272 ;

发展历程不一样 , WM 的手机一开始作为商务机 , 分辨率就是 320×240 , 后边提高到 640×480 , 再到后来的 800×480 。 HVGA 没有作为 WM 手机的重点 , 或者说基本忽略了在这个分辨率级别。

再谈谈 android , 我个人认为安卓是效仿 iPhone 的设计思路。

1 , HVGA 基本上属于性价比比较好的方案。 分辨率适中 , 不像 320×240 的颗粒感那样严重。 因为如果提高分辨率就要提高机器的图像处理组件 , 也就是对硬件的要求相应提高 , 价格自然是不低。

2 , 谷歌手机一开始推广就是用这个分辨率 , 例如 G1 , G2 等都是采用 HVGA 的分辨率。

3 , 现在 android 的软件基本上都会自动适应 HVGA (还包含别的 , 例如 VGA, WVGA 等 , 这个就要看软件作者的了)

1.62 Android 2.2完全退出程序, 使用广播机制

发表时间: 2011-07-15

这个问题一直困扰着我, 试了n种方式, 都不行, 在网上搜了一圈, 没有能用的:

1. ActivityManager manager = (ActivityManager)

context.getSystemService(Context.ACTIVITY_SERVICE);

manager.killBackgroundProcesses(package);

不行

2. android.os.Process.killProcess(android.os.Process.myPid());

也不行

3. manager.restartPackage(package);

还是不行

4.

Intent MyIntent = new Intent(Intent.ACTION_MAIN);

MyIntent.addCategory(Intent.CATEGORY_HOME);

startActivity(MyIntent);

finish();

这个只是退回到桌面, 如果打开多个Activity关闭重新打开也会有问题, 还是不行

看到有说广播机制, 发现是个好东东, 能彻底解决这个问题, 废话不说看代码:

```
public abstract class EnterActivity extends BaseActivity {  
    ...  
    // 写一个广播的内部类, 当收到动作时, 结束activity  
    private BroadcastReceiver broadcastReceiver = new BroadcastReceiver() {  
        @Override  
        public void onReceive(Context context, Intent intent) {  
            close();  
            unregisterReceiver(this); // 这句话必须要写要不然报错, 不写虽然能关闭, 会报  
        }  
    };
```

```
@Override
public void onResume() {
    super.onResume();

    // 在当前的activity中注册广播
    IntentFilter filter = new IntentFilter();
    filter.addAction(Attribute.PAGENAME);
    registerReceiver(this.broadcastReceiver, filter); // 注册
}

/**
 * 关闭
 */
public void close() {
    Intent intent = new Intent();
    intent.setAction(Attribute.PAGENAME); // 说明动作
    sendBroadcast(intent); // 该函数用于发送广播
    finish();
}

...
}
```

别人写的缺了个地方(unregisterReceiver)这个必须要加，找了个最简单的方法

1.63 Android 拖拽图片

发表时间: 2011-07-16

拖拽图片效果

```
package com.iaiai.activity;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnTouchListener;
import android.widget.ImageView;

/**
 *
 * <p>
 * Title: IaiaiActivity.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-6-26
 * </p>
 *
 * @author 丸子
 * @version 0.0.1
 */
```

```
public class IaiaiActivity extends Activity {

    ImageView img;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        img = (ImageView) findViewById(R.id.img_view);

        img.setOnTouchListener(new OnTouchListener() {

            private int mx, my;

            public boolean onTouch(View v, MotionEvent event) {
                switch (event.getAction()) {
                    case MotionEvent.ACTION_MOVE:
                        mx = (int) (event.getRawX());
                        my = (int) (event.getRawY() - 50);

                        v.layout(mx - img.getWidth() / 2, my - img.getHeight()
                                mx + img.getWidth() / 2, my + img.getHeight());
                        Log.i("*****", "Move");
                        break;
                    case MotionEvent.ACTION_DOWN:
                        Log.i("*****", "Down");
                        break;
                    case MotionEvent.ACTION_UP:
                        Log.i("*****", "Up");
                        break;
                }
                return true;
            }
        });
    }
}
```

1.64 Android控件之Gallery实现图片列表

发表时间: 2011-07-18

先来看下运行效果:



费话不多说，直接看代码.....

values/attrs.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <declare-styleable name="Gallery">
        <attr name="android:galleryItemBackground" />
    </declare-styleable>
</resources>
```

layout/img.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Gallery android:id="@+id/galley" android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:layout_x="12px"
        android:layout_y="106px" />
    <ImageView android:id="@+id/image" android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
</LinearLayout>
```

ImageAdapter.java:

```
import java.util.List;

import org.mdx.R;

import android.content.Context;
import android.content.res.TypedArray;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Gallery;
import android.widget.ImageView;

/**
 *
 * <p>
 * Title: ImageAdapter.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
```

```
* <p>
* QQ: 176291935
* </p>
* <p>
* Http: iaiai.iteye.com
* </p>
* <p>
* Create time: 2011-7-18
* </p>
*
* @author 丸子
* @version 0.0.1
*/
public class ImageAdapter extends BaseAdapter {

    int mGalleryItemBackground;
    private Context mContext;
    private List<Integer> list;

    public ImageAdapter(Context c, List<Integer> li) {
        mContext = c;
        list = li;
        TypedArray a = c.obtainStyledAttributes(R.styleable.Gallery);
        mGalleryItemBackground = a.getResourceId(
            R.styleable.Gallery_android_galleryItemBackground, 0);
        a.recycle();
    }

    public int getCount() {
        return list.size();
    }

    public Object getItem(int position) {
        return list.get(position%list.size()).intValue();
    }

    public long getItemId(int arg0) {
```

```
    // TODO Auto-generated method stub
    return arg0;
}

public View getView(int position, View arg1, ViewGroup arg2) {
    // TODO Auto-generated method stub
    ImageView i = new ImageView(mContext);
    Bitmap bm = BitmapFactory.decodeResource(mContext.getResources(), list
        .get(position%list.size()).intValue());
    i.setImageBitmap(bm);
    i.setScaleType(ImageView.ScaleType.FIT_XY);
    i.setLayoutParams(new Gallery.LayoutParams(136, 88));
    i.setBackgroundResource(mGalleryItemBackground);
    return i;
}

}
```

ImgActivity.java:

```
import java.util.ArrayList;
import java.util.List;

import org.mdx.R;
import org.mdx.core.activity.adapter.ImageAdapter;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.Gallery;
import android.widget.ImageView;
```

```
/**  
 *  
 * <p>  
 * Title: ImgActivity.java  
 * </p>  
 * <p>  
 * E-Mail: 176291935@qq.com  
 * </p>  
 * <p>  
 * QQ: 176291935  
 * </p>  
 * <p>  
 * Http: iaiai.iteye.com  
 * </p>  
 * <p>  
 * Create time: 2011-7-18  
 * </p>  
 *  
 * @author 丸子  
 * @version 0.0.1  
 */  
  
public class ImgActivity extends Activity {  
    private ImageView imageView;  
  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.img);  
  
        imageView = (ImageView) findViewById(R.id.image);  
        final Gallery g = (Gallery) findViewById(R.id.galley);  
        final ImageAdapter imageAdapter = new ImageAdapter(this, getImages());  
        g.setAdapter(imageAdapter);  
  
        g.setOnItemClickListener(new OnItemClickListener() {  
            public void onItemClick(AdapterView<?> parent, View view,  
            int position) {  
                Intent intent = new Intent("com.android.camera.action.CAPTURE");  
                intent.putExtra("com.android.camera.action.CAPTURE", position);  
                startActivityForResult(intent, 0);  
            }  
        });  
    }  
}
```

```
        int position, long id) {
    imageView.setImageResource(((Integer) imageAdapter
        .getItem(position)).intValue());
}
});

g.setOnItemSelectedListener(new OnItemSelectedListener() {
    public void onItemSelected(AdapterView<?> parent, View view,
        int position, long id) {
    imageView.setImageResource(((Integer) imageAdapter
        .getItem(position)).intValue());
}

    public void onNothingSelected(AdapterView<?> parent) {
    imageView.setImageResource(((Integer) imageAdapter.getItem(g
        .getSelectedItemPosition())).intValue());
}
});

private List<Integer> getImages() {
    List<Integer> list = new ArrayList<Integer>();
    list.add(new Integer(R.drawable.a));
    list.add(new Integer(R.drawable.b));
    list.add(new Integer(R.drawable.c));
    list.add(new Integer(R.drawable.d));
    list.add(new Integer(R.drawable.e));
    list.add(new Integer(R.drawable.f));
    list.add(new Integer(R.drawable.g));
    return list;
}
}
```

再放几张136*88的图片就可以了

1.65 Android XML文件使用

发表时间: 2011-07-19

一、布局文件：在layout目录下，使用比较广泛；

我们可以为应用定义两套或多套布局，例如：可以新建目录layout_land(代表手机横屏布局)，layout_port(代表手机竖屏布局)，系统会根据不同情况自动找到最合适的布局文件，但是在同一界面的两套不同布局文件的文件名应该是相同的，只是放在了两个不同的目录下。

二、图片文件：在drawable目录下，从2.1版本以后分为三个目录

- drawable-hdpi里面存放高分辨率的图片,如WVGA (480×800),FWVGA (480×854)
- drawable-mdpi里面存放中等分辨率的图片,如HVGA (320×480)
- drawable-ldpi里面存放低分辨率的图片,如QVGA (240×320)

系统会根据机器的分辨率来分别到这几个文件夹里面去找对应的图片。

在开发程序时为了兼容不同平台不同屏幕，建议各自文件夹根据需求均存放不同版本图片。

我们可以将已经做好的图片放到该目录下，或者通过自定义XML文件来实现想要的图片，例如我们可以定义shape_1.xml放到drawable目录下，内容如下：

```
<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="oval">
    <!--android:shape="oval"表示所要绘制的图形是一个椭圆，默认是rectangle，长方形-->
    <gradient
        android:startColor="#0055ff88"
        android:centerColor="#0055ff00"
        android:centerY="0.75"
        android:endColor="#00320077"
        android:angle="270"
    />
    <!--gradient 产生颜色渐变 android:angle 从哪个角度开始变 只有90的整数倍可以 -->
    <solid android:color="#ff4100ff"/>
    <!--solid表示图形是实心的，填充里面，#ff4100ff为填充颜色-->
    <stroke
        android:width="2dp"
        android:color="#ee31ff5e"
        android:dashWidth="3dp"
    />
```

```
    android:dashGap="2dp" />
<!-- 描边 采用那样的方式将外形轮廓线画出来 , width表示笔的粗细 , dashWidth表示小横线的宽度 , dashGap表示
<padding
    android:left="7dp"
    android:top="7dp"
    android:right="7dp"
    android:bottom="7dp" />
<!--和CSS中的padding应该是一个道理-->
<corners android:radius="6dp" />
<!--corners表示是有半径为5像素的圆角-->
</shape>
```

当我们想让一个控件根据不同状态显示不同图片，可以直接在程序中控制，也可以在drawable目录建立XML文件达到相同的效果，例如：我们可以在drawable目录下新建文件button_back.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_pressed="false" android:drawable="@drawable/xxx1" />
    <item android:state_pressed="true" android:drawable="@drawable/xxx2" />
    <item android:state_focused="true" android:drawable="@drawable/xxx3" />
    <!-- 这里还可以加N多效果和动作 只要你用的到 -->
    <item android:drawable="@drawable/xxx4" />
</selector>
```

以上XML文件可以实现一个控件（假设为button），获取焦点，按下按钮，正常状态下显示不同图片的效果，只需要在定义控件时引用该文件名即可，例如：

```
<Button
    android:id="@+id/Button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/button_add_x">
```

```
</Button>  
<!--android:background="@drawable/button_back"指向button_back.xml文件-->
```

但是当我们的条件不是系统已有的事件类型，例如根据ImageView根据一个变量var的值的不同显示不同的图片，应该怎么办呢？可以在程序中写如下代码

```
if (条件1){  
    image.setBackground(R.id.xxx1);  
}else if (条件2){  
    image.setBackground(R.id.xxx2);  
} ...
```

或者可以用另一个简便的方法实现相同的功能，在res/drawable下建立一个xml文件，内容如下

```
<level-list xmlns:android="http://schemas.android.com/apk/res/android">  
    <item android:maxLevel="9" android:drawable="@drawable/battery_0" />  
    <item android:maxLevel="39" android:drawable="@drawable/battery_1" />  
    <item android:maxLevel="69" android:drawable="@drawable/battery_2" />  
    <item android:maxLevel="89" android:drawable="@drawable/battery_3" />  
    <item android:maxLevel="100" android:drawable="@drawable/battery_4" />  
</level-list>
```

然后在layout中把imageview的src设置成已创建好的xml文件，程序中变换图片时，只需要使用imageview.getDrawable().setLevel(50);Android会根据level的值自动选择对应的图片。手机显示剩余电量就是用这个方法来显示不同图片的。

三、菜单文件：在menu目录下，写代码时只需在onCreateOptionsMenu方法中用MenuInflater装载进去就OK了。格式如下

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
    <item android:id="@+id/enabled_item"  
        android:title="Enabled"  
        android:icon="@drawable/stat_happy" />
```

```
<item android:id="@+id/disabled_item"
    android:title="Disabled"
    android:enabled="false"
    android:icon="@drawable/stat_sad" />

<item android:id="@+id/enabled_item_2"
    android:title="Enabled"
    android:icon="@drawable/stat_happy" />

<item android:id="@+id/disabled_item_2"
    android:title="Disabled"
    android:enabled="false"
    android:icon="@drawable/stat_sad" />

</menu>
```

四、resource文件，在values目录下，之所以称之为resource文件，是因为values目录下xml文件都是以resource作为根节点

1.strings.xml 定义字符串的文件，格式如下：

```
<resources>
    <string name="hello">Hello World!</string>
    <string name="app_name">我的应用程序</string>
</resources>
```

2.colors.xml 定义颜色的文件，格式如下：

```
<resources>
    <!--定义图片颜色-->
    <drawable name="screen_background_black">#ff000000</drawable>
    <drawable name="translucent_background">#e0000000</drawable>
    <drawable name="transparent_background">#00000000</drawable>
```

```
<!--定义文字颜色-->  
<color name="solid_red">#f00</color>  
<color name="solid_blue">#0000ff</color>  
<color name="solid_green">#f0f0</color>  
<color name="solid_yellow">#fffff00</color>  
</resources>
```

3.array.xml 定义数组的文件，格式如下：

```
<resources>  
    <string-array name="planets">  
        <item>Mercury</item>  
        <item>Venus</item>  
        <item>Earth</item>  
        <item>Mars</item>  
        <item>Jupiter</item>  
        <item>Saturn</item>  
        <item>Uranus</item>  
        <item>Neptune</item>  
        <item>Pluto</item>  
    </string-array>  
  
    <integer-array name="numbers">  
        <item>100</item>  
        <item>500</item>  
        <item>800</item>  
    </integer-array>  
</resources>
```

4.styles.xml 定义样式的文件，分为两种用途：

Style：以一个单位的方式用在布局XML单个元素（控件）当中。例如：我们可以为TextView定义一种样式风格，包含文本的字号大小和颜色，然后将其用在TextView特定的实例。

Theme：以一个单位的方式用在应用中所有的Activity当中或者应用中的某个Activity当中。比如，我们可以定义一个Theme，它为window frame和panel 的前景和背景定义了一组颜色，并为菜单定义可文字的大小和颜色属性，可以将这个Theme应用在你程序当中所有的Activity里。

```
<resources>
    <!--Theme,可以用来定义activity的主题-->
    <style name="Theme.Transparent">
        <item name="android:windowIsTranslucent">true</item>
        <item name="android:windowAnimationStyle">@android:style/Animation.Translucent</item>
        <item name="android:windowBackground">@drawable/transparent_background</item>
        <item name="android:windowNoTitle">true</item>
        <item name="android:colorForeground">#fff</item>
    </style>
    <!--Style,可以用来定义某个View元素，这里是ImageView的样式-->
    <style name="ImageView120dpi">
        <item name="android:src">@drawable/stylogo120dpi</item>
        <item name="android:layout_width">wrap_content</item>
        <item name="android:layout_height">wrap_content</item>
    </style>
</resources>
```

个人认为，其实不管是Theme还是Style，其实只是应用的范围不同而已，区分的话应该是根据 android:name="xxxx"的xxxx来区分吧，很明显是不同的。

5.dimen.xml 定义单位的文件，android中度量单位有以下几种：

- px(象素)：屏幕实际的象素，常说的分辨率1024*768pixels，就是横向1024px, 纵向768px，不同设备显示效果相同。
- in(英寸)：屏幕的物理尺寸，每英寸等于2.54厘米。
- mm(毫米)：屏幕的物理尺寸。
- pt(点)：屏幕的物理尺寸。1/72英寸。
- dp/dip：与密度无关的象素，一种基于屏幕密度的抽象单位。在每英寸160点的显示器上，1dp = 1px。但dp和px的比例会随着屏幕密度的变化而改变，不同设备有不同的显示效果。
- sp：与刻度无关的象素，主要用于字体显示best for textsize，作为和文字相关大小单位。

```
<resources>
    <dimen name="one_pixel">1px</dimen>
    <dimen name="double_density">2dp</dimen>
    <dimen name="sixteen_sp">16sp</dimen>
</resources>
```

6.attrs.xml 定义属性的文件，主要用在自定义的组件中，具体使用方法会在后续的如何使用自定义组件中详细介绍，其格式如下：

```
<resources>
    <declare-styleable name="MyView">
        <attr name="textColor" format="color" />
        <attr name="textSize" format="dimension" />
    </declare-styleable>
</resources>
```

五、动画文件 在anim目录下，动画资源分为两种

1.实现图片的translate、scale、rotate、alpha四种变化，还可以设置动画的播放特性，称为Tween动画。

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate android:interpolator="@android:anim/accelerate_interpolator"
        android:fromXDelta="0" android:toXDelta="200" android:fromYDelta="0"
        android:toYDelta="180" android:duration="2000" />
    <scale android:interpolator="@android:anim/accelerate_interpolator"
        android:fromXScale="1.0" android:toXScale="2.0" android:fromYScale="1.0"
        android:toYScale="2.0" android:pivotX="150%" android:pivotY="150%"
        android:duration="2000" />
    <alpha android:fromAlpha="1.0" android:toAlpha="1.0"
        android:duration="@android:integer/config_mediumAnimTime" />
    <rotate ....各个属性></rotate>
    <Interpolator>可以使用其子类和属性定义动画的运行方式，先快后慢，先慢后快等</Interpolator>
</set>
```

2.帧动画，逐帧播放设置的资源，称为Frame动画。

```
<animation-list xmlns:android=" http://schemas.android.com/apk/res/android"
    android:oneshot=" true" >
    <item android:drawable=" @drawable/rocket_thrust1" android:duration=" 200" />
    <item android:drawable=" @drawable/rocket_thrust2" android:duration=" 200" />
    <item android:drawable=" @drawable/rocket_thrust3" android:duration=" 200" />
</animation-list>
```

六、raw目录下的文件，是直接复制到设备中的任意文件。它们无需编译，添加到你的应用程序编译产生的压缩文件中。一般为应用要用到的音频或视频文件等等

要使用这些资源，可以调用Resources.openRawResource()，参数是资源的ID，即R.raw.somefilename。

七、xml目录下的文件，是程序中需要使用的普通xml文件。在运行时可以通过调用Resources.getXML()读取。

八、assets目录下的文件都是保持原始的文件格式，需要用AssetManager以字节流的形式读取文件。

- 先在Activity里面调用getAssets()来获取AssetManager引用。
- 再用AssetManager的open(String fileName, int accessMode)方法则指定读取的文件以及访问模式就能得到输入流InputStream。
- 然后就是用已经open file 的InputStream读取文件，读取完成后记得InputStream.close()。
- 调用AssetManager.close()关闭AssetManager。

总结：其实android中定义如此多的XML配置文件，在我看来就是为了达到显示层和数据层的分离，提高了可维护性，也是我们的程序代码变得简洁。

1.66 Android中尺寸单位杂谈

发表时间: 2011-07-19

在android系统中单位DP也就是DIP : device independent pixels(设备独立像素).

- px (Pixels , 像素) : 对应屏幕上的实际像素点。
- in (Inches , 英寸) : 屏幕物理长度单位。
- mm (Millimeters , 毫米) : 屏幕物理长度单位。
- pt (Points , 磅) : 屏幕物理长度单位 , 1/72英寸。
- dp (与密度无关的像素) : 逻辑长度单位 , 在 160 dpi 屏幕上 , $1dp=1px=1/160$ 英寸。随着密度变化 , 对应的像素数量也变化 , 但并没有直接的变化比例。
- dip : 与dp相同 , 多用于Google示例中。
- sp (与密度和字体缩放度无关的像素) : 与dp类似 , 但是可以根据用户的字体大小首选项进行缩放。

尽量使用dp作为空间大小单位 , sp作为和文字相关大小单位

dp也就是dip。这个和sp基本类似。如果设置表示长度、高度等属性时可以使用dp或sp。但如果设置字体 , 需要使用sp。dp是与密度无关 , sp除了与密度无关外 , 还与scale无关。如果屏幕密度为160 , 这时dp和sp和px是一样的。 $1dp=1sp=1px$, 但如果使用px作单位 , 如果屏幕大小不变 (假设还是3.2寸) , 而屏幕密度变成了320。那么原来TextView的宽度设成160px , 在密度为320的3.2寸屏幕里看要比在密度为160的3.2寸屏幕上短了一半。但如果设置成160dp或160sp的话。系统会自动将 width属性值设置成320px的。也就是 $160*320/160$ 。其中 $320/160$ 可称为密度比例因子。也就是说 , 如果使用dp和sp , 系统会根据屏幕密度的变化自动进行转换。

使用尺寸 (dimen) 资源

我们可以使用一些常用的尺寸单位来定义一些文字尺寸、视图组件的宽和高等。尺寸资源是一个数字类型的数据 , 被定义在res/values/dimens.xml文件中。

Android中支持的尺寸单位

在计算机中一般我们会用到的尺寸单位有厘米 (cm) 、毫米 (mm) 、像素 (px) 、英尺 (in) 等。Android中支持的尺寸单位如表3.4所示。

表 Android中支持单位

单位	表示单位	名称	单位说明
px	像素		屏幕上的真实像素表示

in	英尺	基于屏幕的物理尺寸
mm	毫米	基于屏幕的物理尺寸
pt	点	英尺的1/72
dp	和密度无关的像素	相对屏幕物理密度的抽象单位
sp	和精度无关的像素	和dp类似

尺寸资源XML文件的定义

尺寸资源是定义在XML文件中的一些整型数值。有关尺寸资源的定义如表

资源位置 res/values/dimens.xml

尺寸XML文件格式 使用<?xml version="1.0" encoding="utf-8"?><resources>根元素<dimen>子元素：
<dimen name=dimen_name>dimen_value</dimen>

获得尺寸资源的方法 getResources().getDimension()

引用尺寸资源的格式 Java代码中：R.dimen.dimen_name----XML文件中：@[package:]dimen/dimen_name

1.67 Android系统中的.apk文件和dex文件

发表时间: 2011-07-19

1.apk文件

APK是Android Package的缩写，即Android安装包。APK是类似Symbian Sis或Sisx的文件格式，PC上的安装文件的后缀名为“.exe”。通过将APK文件直接传到Android模拟器或Android手机中执行即可安装。

使用Android打包工具(aapt)将DEX文件，资源文件以及AndroidManifest.xml文件(二进制格式)组合成一个应用程序包(APK)。APK文件其实是zip格式，但后缀名被修改为apk，通过UnZip解压后，可以看到Dex文件，Dex是Dalvik VM executes的全称，即Android Dalvik执行程序，并非Java ME的字节码而是Dalvik字节码。

一个APK文件结构为：

META-INF\ Jar文件中常可以看到

res\ 存放资源文件的目录

AndroidManifest.xml 程序全局配置文件

classes.dex Dalvik字节码

resources.arsc 编译后的二进制资源文件

总结下我们发现Android在运行一个程序时首先需要UnZip，然后类似Symbian那样直接，和Windows Mobile中的PE文件有区别，这样做对于程序的保密性和可靠性不是很高，通过dexdump命令可以反编译，但这样做符合发展规律，微软的Windows Gadgets或者说WPF也采用了这种构架方式。

在Android平台中dalvik vm的执行文件被打包为apk格式，最终运行时加载器会解压然后获取编译后的androidmanifest.xml文件中的permission分支相关的安全访问，但仍然存在很多安全限制，如果你将apk文件传到/system/app文件夹下会发现执行是不受限制的。最终我们平时安装的文件可能不是这个文件夹，而在android rom中系统的apk文件默认会放入这个文件夹，它们拥有着root权限。

2.dex文件

Dex是Dalvik VM executes的全称，即Android Dalvik执行程序，并非Java ME的字节码而是Dalvik字节码。Google在新发布的Android平台上使用了自己的Dalvik虚拟机来定义，这种虚拟机执行的并非Java字节码，而是另一种字节码：dex格式的字节码。在编译Java代码之后，通过Android平台上的工具可以将Java字节码转换成Dex字节码。虽然Google称Dalvik是为了移动设备定做的，但是业界很多人认为这是为了规避向sun申请Java license。这个Dalvik VM针对手机程序程式/CPU做过最佳化，可以同时执行许多VM而不会占用太多Resource。

对于Android DEX文件进行优化，需要注意的一点是DEX文件的结构是紧凑的，但是我们还是要想方设法的进行提高程序的运行速度，我们就仍然需要对DEX文件进行进一步优化。

调整所有字段的字节序（LITTLE_ENDIAN）和对齐结构中的每一个域 验证DEX文件中的所有类 对一些特定的类进行优化，对方法里的操作码进行优化。优化后的文件大小会有所增加，应该是原Android DEX文件的1-4倍。优化发生的时机有两个：对于预置应用，可以在系统编译后，生成优化文件，以ODEX结尾。

这样在发布时除APK文件（不包含DEX）以外，还有一个相应的Android DEX文件；对于非预置应用，包含在APK文件里的DEX文件会在运行时被优化，优化后的文件将被保存在缓存中



每一个Android应用都运行在一个Dalvik虚拟机实例里，而每一个虚拟机实例都是一个独立的进程空间。虚拟机的线程机制，内存分配和管理，Mutex等等都是依赖底层操作系统而实现的。

所有Android应用的线程都对应一个Linux线程，虚拟机因而可以更多的依赖操作系统的线程调度和管理机制。不同的应用在不同的进程空间里运行，加之对不同来源的应用都使用不同的Linux用户来运行，可以最大程度的

保护应用的安全和独立运行。

Zygote是一个虚拟机进程，同时也是一个虚拟机实例的孵化器，每当系统要求执行一个Android应用程序，Zygote就会FORK出一个子进程来执行该应用程序。这样做的好处显而易见：Zygote进程是在系统启动时产生的，它会完成虚拟机的初始化，库的加载，预置类库的加载和初始化等等操作，而在系统需要一个新的虚拟机实例时。

Zygote通过复制自身，最快速的提供个系统。另外，对于一些只读的系统库，所有虚拟机实例都和Zygote共享一块内存区域，大大节省了内存开销。Android应用开发和Dalvik虚拟机Android应用所使用的编程语言是Java语言，和Java SE一样，编译时使用Sun JDK将Java源程序编程成标准的Java字节码文件（.class文件）。

而后通过工具软件DX把所有的字节码文件转成Android DEX文件（classes.dex）。最后使用Android打包工具（aapt）将DEX文件，资源文件以及AndroidManifest.xml文件（二进制格式）组合成一个应用程序包（APK）。应用程序包可以被发布到手机上运行。

1.68 Android 调用系统摄像头

发表时间: 2011-07-21

在开发android 应用的时候我们经常需要用到摄像头来进行拍照或者是录制视频，这里介绍一种最简单的方法：调用系统摄像头来拍照，并将照片保存起来

启动摄像头的方法：

```
startActivityForResult(new Intent("android.media.action.IMAGE_CAPTURE"), TAKE_PICTURE);
```

拍摄成功后接收参数的接收方法我们只需要重写

protected void onActivityResult(int requestCode, int resultCode, Intent data)方法即可

如：

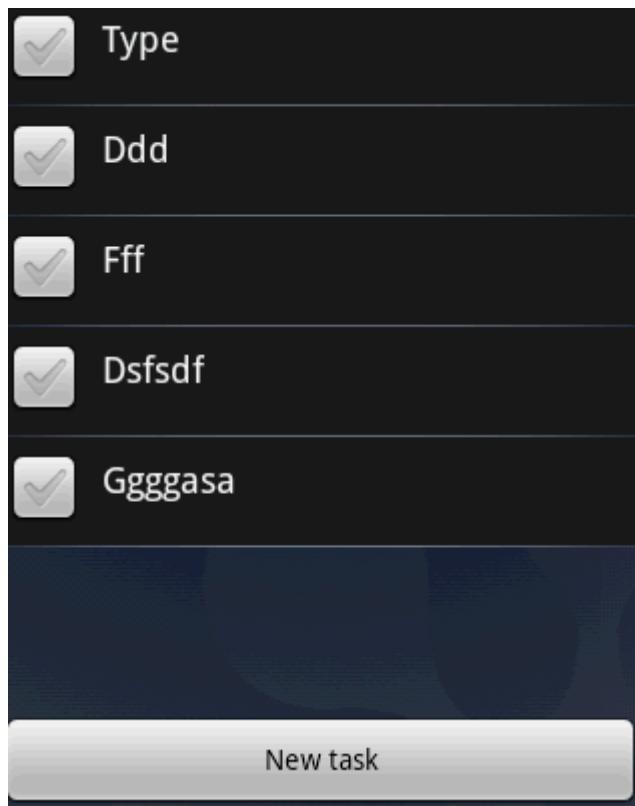
```
final int TAKE_PICTURE = 1;
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == TAKE_PICTURE) {
        if (resultCode == RESULT_OK) {
            Bitmap bm = (Bitmap) data.getExtras().get("data");
            imgPhoto.setImageBitmap(bm);
            sdCardExit = Environment.getExternalStorageState().equals(android.os.Environment.MEDIA_MOUNTED);
        }
        if (sdCardExit) {
            myRecAudioDir = new File(Environment.getExternalStorageDirectory().getAbsolutePath() + '我的录音');
            if (!myRecAudioDir.exists()) {
                myRecAudioDir.mkdir();
            }
        } else {
            mMakeTextToast("", true);
        }
    }
    try {
        File f = File.createTempFile(strTempFile, ".jpg", myRecAudioDir);
    }
}
```


1.69 Android 自定义ListView背景

发表时间: 2011-07-22

在Android中，ListView是最常用的一个控件，在做UI设计的时候，很多人希望能够改变一下它的背景，使他能够符合整体的UI设计，改变背景很简单只需要准备一张图片然后指定属性

android:background="@drawable/bg"，不过不要高兴地太早，当你这么做以后，发现背景是变了，但是当你拖动，或者点击list空白位置的时候发现ListItem都变成黑色的了，破坏了整体效果，如下图所示

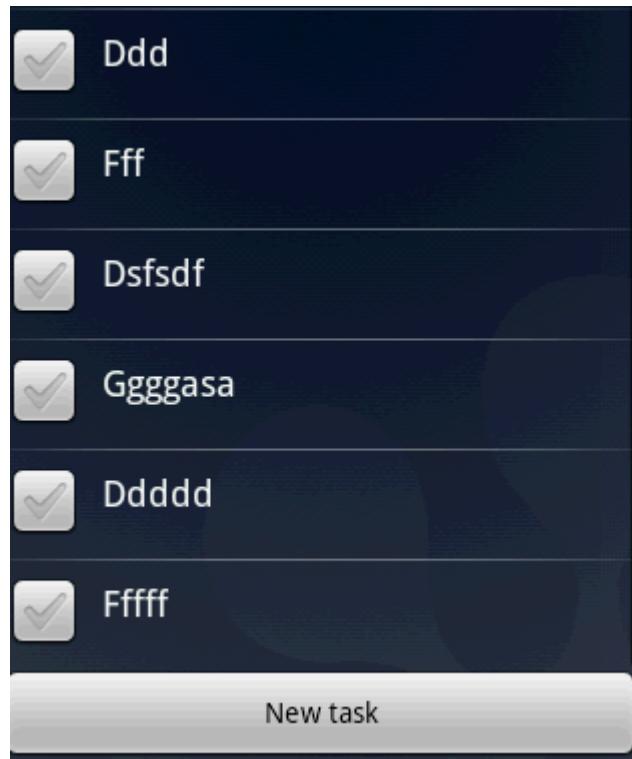


这是为什么呢？

这个要从Listview的效果说起，默认的ListItem背景是透明的，而ListView的背景是固定不变的，所以在滚动条滚动的过程中如果实时地去将当前每个Item的显示内容跟背景进行混合运算，所以android系统为了优化这个过程用，就使用了一个叫做**android:cacheColorHint**的属性，在黑色主题下默认的颜色值是#191919，所以就出现了刚才的画面，有一半是黑色的

那怎么办呢？

如果你只是换背景的颜色的话，可以直接指定**android:cacheColorHint**为你所要的颜色，如果你是用图片做背景的话，那也只要将**android:cacheColorHint**指定为透明 (#00000000) 就可以了，当然为了美化是要牺牲一些效率的。最后美化的效果如图：

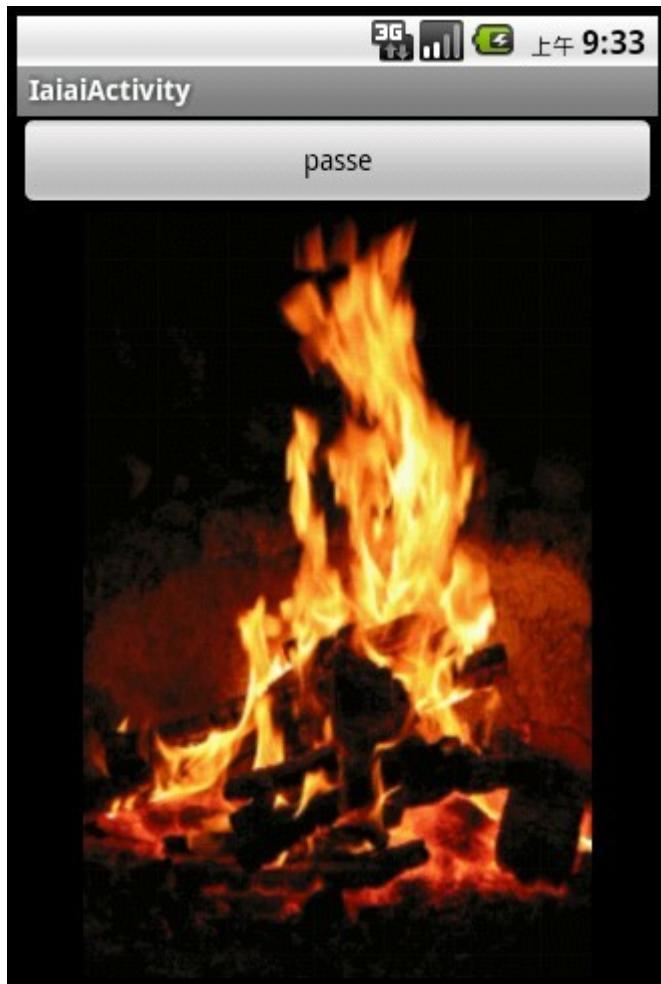


1.70 Android frame by frame animation动画显示

发表时间: 2011-07-28

在看到编写简单的动画的时候，想到了android上也可以做到这一点，只是几个图片来回的切换。这种显示方式学名叫做：frame by frame animation，顺序播放事先做好的图像，跟电影类似；

效果：



res/anim/picture_animation.xml:

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

<!-- 动画帧集合对象 --&gt;
&lt;animation-list xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot="false"&gt;
    &lt;!--动画帧对象 android:duration表示每帧动画显示的时间,放在drawable下的动画图片不能太大,否则会卡住--&gt;
    &lt;item android:drawable="@drawable/camp_fire1" android:duration="83" /&gt;
    &lt;item android:drawable="@drawable/camp_fire2" android:duration="83" /&gt;
    &lt;item android:drawable="@drawable/camp_fire3" android:duration="83" /&gt;</pre>
```

```
<item android:drawable="@drawable/camp_fire4" android:duration="83" />
<item android:drawable="@drawable/camp_fire5" android:duration="83" />
<item android:drawable="@drawable/camp_fire6" android:duration="83" />
<item android:drawable="@drawable/camp_fire7" android:duration="83" />
<item android:drawable="@drawable/camp_fire8" android:duration="83" />
<item android:drawable="@drawable/camp_fire9" android:duration="83" />
<item android:drawable="@drawable/camp_fire10" android:duration="83" />
<item android:drawable="@drawable/camp_fire11" android:duration="83" />
<item android:drawable="@drawable/camp_fire12" android:duration="83" />
<item android:drawable="@drawable/camp_fire13" android:duration="83" />
<item android:drawable="@drawable/camp_fire14" android:duration="83" />
<item android:drawable="@drawable/camp_fire15" android:duration="83" />
<item android:drawable="@drawable/camp_fire16" android:duration="83" />
<item android:drawable="@drawable/camp_fire17" android:duration="83" />
</animation-list>
```

main.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Button android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:text="start" android:id="@+id/start"
    <ImageView android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:id="@+id/imageId"
        android:src="@anim/picture_animation" />
</LinearLayout>
```

IaiaiActivity.java类:

```
package com.iaiai.activity;
```

```
import android.app.Activity;
import android.graphics.drawable.AnimationDrawable;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;

/**
 *
 * <p>
 * Title: IaiaiActivity.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-7-28
 * </p>
 *
 * @author 丸子
 * @version 0.0.1
 */
public class IaiaiActivity extends Activity {

    private ImageView imageView;

    private AnimationDrawable draw = null;
    private Button start;

    @Override
```

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
    this.imageView = (ImageView) findViewById(R.id.imageId);  
    this.draw = (AnimationDrawable) imageView.getDrawable();  
  
    this.start = (Button) findViewById(R.id.start);  
    this.start.setOnClickListener(new OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            if (draw.isRunning()) {  
                draw.stop();  
                start.setText("start");  
            } else {  
                draw.stop();  
                draw.start();  
                start.setText("passe");  
            }  
        }  
    });  
  
}  
}
```

注:代码运行的结果想必大家应该就知道了(图片按照顺序的播放一次),不过有一点需要强调的是:启动Frame Animation动画的代码rocketAnimation.start();不能在OnCreate()中,因为在OnCreate()中AnimationDrawable还没有完全的与ImageView绑定,在OnCreate()中启动动画,就只能看到第一张图片。

XML属性:

属性	说明
drawable	当前帧引用的drawable资源
duration	当前帧显示的时间(毫秒为单位)
oneshot	如果为true,表示动画只播放一次停止在最后一帧上,如果设置为false表示动画循环播放。
variablePadding	If true, allows the drawable's padding to change based on the current state that is selected.

visible 规定drawable的初始可见性，默认为false；

下面，阅读Android SDK中对AnimationDrawable的介绍，有个简单的了解：

AnimationDrawable

获取、设置动画的属性

int getDuration() 获取动画的时长

int getNumberOfFrames() 获取动画的帧数

boolean isOneShot()/Void setOneShot(boolean oneshot) 获取oneshot属性/设置oneshot属性

void inflate(Resource r,XmlPullParser p,AttributeSet attrs)

增加、获取帧动画

Drawable getFrame(int index) 获取某帧的Drawable资源

void addFrame(Drawable frame,int duration) 为当前动画增加帧 (资源，持续时长)

动画控制

void start() 开始动画

void run() 外界不能直接掉调用，使用start()替代

boolean isRunning() 当前动画是否在运行

void stop() 停止当前动画

附件下载：

- [donghua.rar \(3.9 MB\)](#)
- dl.iteye.com/topics/download/563ef10d-2613-31f8-b430-78f6a14d3e2c

1.71 Android HandlerThread类 , Bundle的使用

发表时间: 2011-07-29

1、其实上面的方法 , 直接使用handler.post(updateThread),然后线程updateThread直接调用run()方法 , 并没有调用start()方法 , 所以并没有产生新的线程 , 都是在主线程里面运行的

2、要产生新的线程 , 可以用如下方法 :

```
//生成一个HandlerThread对象 , 实现了使用Looper来处理消息队列的功能 , 这个类由Android应用程序框架提供
HandlerThread handlerThread = new HandlerThread("handler_thread");
//必须先调用该类的start();
handlerThread.start();
```

这样就产生了新的线程 , 然后可以继承Handler类 , 生成Handler对象 , 来控制线程

3、Bundle是键值对 , 键为string类型 , 值可以为多种类型 , 用于Message的setData()中存储数据的 , 加入数据如下

```
Bundle b = new Bundle();
b.putInt("age", 20);
b.putString("name", "Jhon");
```

```
package mars.handler;

import android.app.Activity;
import android.os.Bundle;
import android.os.Handler;
import android.os.HandlerThread;
import android.os.Looper;
import android.os.Message;

public class HandlerTest2 extends Activity {
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    HandlerThread handlerThread = new HandlerThread("handler_thread");
    handlerThread.start();

    MyHandler myHandler = new MyHandler(handlerThread.getLooper());
    Message msg = myHandler.obtainMessage();

    Bundle b = new Bundle();
    b.putInt("age", 20);
    b.putString("name", "Jhon");
    msg.setData(b);
    msg.sendToTarget();
}

//继承Handler类 ,
class MyHandler extends Handler{
    public MyHandler(Looper looper){
        super(looper);
    }
    @Override
    public void handleMessage(Message msg) {
        Bundle b = msg.getData();
        int age = b.getInt("age");
        String name = b.getString("name");
        System.out.println("age is " + age + ", name is" + name);

    }
}
```

此代码在实际运行的过程虽然没有问题，但是如果是想在void handleMessage()方法里刷新View和控件会报如下错误：

```
...:only the original thread that created a view hierarchy can touch its views.
```

这主要是Android的相关View和控件不是线程安全的，我们必须做独立的处理。这里Android给我们提供了很多方法，今天android开发网说一种简单的方法除了异步任务AsyncTask外使用Handler可以很好的处理，和Win32的消息很像。

首先我们需要明白，主线程或者这里说的原始线程original thread 一般情况下是UI线程，当然UI线程并不一定是主线程，我们不能长时间的阻塞该应用，在Android平台上可能会产生类似Force close或Wait这样的对话框这里我们成为ANR，这里除了使用ProgressDialog方式给用一个动态的进度代表当前处理并没有中断可能需要一些时间，所以告诉大家相关的网络处理可以使用工作者线程，但是worker线程不能处理显示元素即UI相关的View或Widget包中的高层的控件，所以通过一个Handler对象可以很好的传递Runnable或Message，下面我们用一个简单的例子来描述

```
final Handler handler = new Handler();
new Thread() {
    public void run() {
        // list = getData();    //处理得到结果了，这里一些内容保存在主类的成员变量中
        handler.post(new Runnable() {
            public void run() {
                //这里就可以获得主类中的组件刷新
                gallery = (Gallery) findViewById(R.id.gallery);
                gallery.setAdapter(new ImageAdapter(MainActivity.this,
                    listData));
                getLoading().dismiss();
            }
        });
    }
}.start();
```

1.72 Android Launcher研究

发表时间: 2011-07-29

费话不多说，直接看附件...

附件下载:

- [Android_Launcher研究.pdf \(1.5 MB\)](#)
- dl.iteye.com/topics/download/dbdd6e3b-c438-34cd-a425-00b43df53927

1.73 Android 图片加圆角以及时间处理工具类

发表时间: 2011-08-03

给图片加上圆角效果好看多了。



加圆角 , Drawable , Bitmap, BitmapDrawable,字节数组之间的相互转换。

```
public class ImageUtil {  
  
    public static InputStream getRequest(String path) throws Exception {  
        URL url = new URL(path);  
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
        conn.setRequestMethod("GET");  
        conn.setConnectTimeout(5000);  
        if (conn.getResponseCode() == 200){  
            return conn.getInputStream();  
        }  
        return null;  
    }  
  
    public static byte[] readInputStream(InputStream inStream) throws Exception {  
        ByteArrayOutputStream outSteam = new ByteArrayOutputStream();  
        byte[] buffer = new byte[4096];  
        int len = 0;  
        while ((len = inStream.read(buffer)) != -1) {  
            outSteam.write(buffer, 0, len);  
        }  
        outSteam.close();  
        inStream.close();  
        return outSteam.toByteArray();  
    }  
  
    public static Drawable loadImageFromUrl(String url){
```

```
URL m;
InputStream i = null;
try {
    m = new URL(url);
    i = (InputStream) m.getContent();
} catch (MalformedURLException e1) {
    e1.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
Drawable d = Drawable.createFromStream(i, "src");
return d;
}

public static Drawable getDrawableFromUrl(String url) throws Exception{
    return Drawable.createFromStream(getRequest(url),null);
}

public static Bitmap getBitmapFromUrl(String url) throws Exception{
    byte[] bytes = getBytesFromUrl(url);
    return byteToBitmap(bytes);
}

public static Bitmap getRoundBitmapFromUrl(String url,int pixels) throws Exception{
    byte[] bytes = getBytesFromUrl(url);
    Bitmap bitmap = byteToBitmap(bytes);
    return toRoundCorner(bitmap, pixels);
}

public static Drawable getRoundDrawableFromUrl(String url,int pixels) throws Exception{
    byte[] bytes = getBytesFromUrl(url);
    BitmapDrawable bitmapDrawable = (BitmapDrawable)byteToDrawable(bytes);
    return toRoundCorner(bitmapDrawable, pixels);
}

public static byte[] getBytesFromUrl(String url) throws Exception{
    return readInputStream(getRequest(url));
```

```
}

public static Bitmap byteToBitmap(byte[] byteArray){
    if(byteArray.length!=0){
        return BitmapFactory.decodeByteArray(byteArray, 0, byteArray.length);
    }
    else {
        return null;
    }
}

public static Drawable byteToDrawable(byte[] byteArray){
    ByteArrayInputStream ins = new ByteArrayInputStream(byteArray);
    return Drawable.createFromStream(ins, null);
}

public static byte[] Bitmap2Bytes(Bitmap bm){
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    bm.compress(Bitmap.CompressFormat.PNG, 100, baos);
    return baos.toByteArray();
}

public static Bitmap drawableToBitmap(Drawable drawable) {
    Bitmap bitmap = Bitmap
        .createBitmap(
            drawable.getIntrinsicWidth(),
            drawable.getIntrinsicHeight(),
            drawable.getOpacity() != PixelFormat.OPAQUE ? Bitmap.Config.ARGB_8888
                : Bitmap.Config.RGB_565);
    Canvas canvas = new Canvas(bitmap);
    drawable.setBounds(0, 0, drawable.getIntrinsicWidth(),
        drawable.getIntrinsicHeight());
    drawable.draw(canvas);
    return bitmap;
}
```

```
/***
 * 图片去色,返回灰度图片
 * @param bmpOriginal 传入的图片
 * @return 去色后的图片
 */

public static Bitmap toGrayscale(Bitmap bmpOriginal) {
    int width, height;
    height = bmpOriginal.getHeight();
    width = bmpOriginal.getWidth();

    Bitmap bmpGrayscale = Bitmap.createBitmap(width, height, Bitmap.Config.RGB_565);
    Canvas c = new Canvas(bmpGrayscale);
    Paint paint = new Paint();
    ColorMatrix cm = new ColorMatrix();
    cm.setSaturation(0);
    ColorMatrixColorFilter f = new ColorMatrixColorFilter(cm);
    paint.setColorFilter(f);
    c.drawBitmap(bmpOriginal, 0, 0, paint);
    return bmpGrayscale;
}

/***
 * 去色同时加圆角
 * @param bmpOriginal 原图
 * @param pixels 圆角弧度
 * @return 修改后的图片
 */

public static Bitmap toGrayscale(Bitmap bmpOriginal, int pixels) {
    return toRoundCorner(toGrayscale(bmpOriginal), pixels);
}

/***
 * 把图片变成圆角
 * @param bitmap 需要修改的图片
 * @param pixels 圆角的弧度
 * @return 圆角图片
 */
```

```
*/  
  
public static Bitmap toRoundCorner(Bitmap bitmap, int pixels) {  
  
    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 = pixels;  
  
    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;  
}  
  
/**  
 * 使圆角功能支持BitmapDrawable  
 * @param bitmapDrawable  
 * @param pixels  
 * @return  
 */  
  
public static BitmapDrawable toRoundCorner(BitmapDrawable bitmapDrawable, int pixels) {  
    Bitmap bitmap = bitmapDrawable.getBitmap();  
    bitmapDrawable = new BitmapDrawable(toRoundCorner(bitmap, pixels));  
    return bitmapDrawable;  
}  
}
```

时间类:

```
public class TimeUtil {  
  
    public static String converTime(long timestamp){  
        long currentSeconds = System.currentTimeMillis()/1000;  
        long timeGap = currentSeconds-timestamp;//与现在时间相差秒数  
        String timeStr = null;  
        if(timeGap>24*60*60){//1天以上  
            timeStr = timeGap/(24*60*60)+"天前";  
        }else if(timeGap>60*60){//1小时-24小时  
            timeStr = timeGap/(60*60)+"小时前";  
        }else if(timeGap>60){//1分钟-59分钟  
            timeStr = timeGap/60+"分钟前";  
        }else{//1秒钟-59秒钟  
            timeStr = "刚刚";  
        }  
        return timeStr;  
    }  
  
    public static String getStandardTime(long timestamp){  
        SimpleDateFormat sdf = new SimpleDateFormat("MM月dd日 HH:mm");  
        Date date = new Date(timestamp*1000);  
        sdf.format(date);  
        return sdf.format(date);  
    }  
}
```

1.74 Android浏览器显示大分辨率图片的问题 分享

发表时间: 2011-08-04

很多人都发现android的浏览器显示大分辨图片会出现显示模糊的问题，我测试是超过100万像素 (1024x1024) 的图片就会出现模糊。在android的讨论网站上也有对这个问题的一个 issue (<http://code.google.com/p/android/issues/detail?id=6021>)。很明显这是android的浏览器实现代码在加载图片的时候做了限制。相关代码在 external/webkit/WebCore/platform/graphics/android /ImageSourceAndroid.cpp，这个文件中有一个局部函数computeMaxBitmapSizeForCache()，看一下这个函数前面的注释：

Images larger than this should be subsampled. Using ashmem, the decoded pixels will be purged as needed, so this value can be pretty large. Making it too small hurts image quality (e.g. abc.com background). 2Meg works for the sites I've tested, but if we hit important sites that need more, we should try increasing it and see if it has negative impact on performance (i.e. we end up thrashing because we need to keep decoding images that have been purged.

Perhaps this value should be some fraction of the available RAM...

已经说得很明白了，可以修改这个函数里面的常数 (2*1024*1024)，不过修改这个值以后记得改一下上面的一个宏定义MIN_RLE_ALLOC_SIZE为同样的值，否则在加载大图片时会很慢。

1.75 Android获取其他包的Context实例然后做坏事

发表时间: 2011-08-04

Android中有Context的概念，想必大家都知道。Context可以做很多事情，打开activity、发送广播、打开本包下文件夹和数据库、获取classLoader、获取资源等等。如果我们得到了一个包的Context对象，那我们基本上可以做这个包自己能做的大部分事情。

那我们能得到吗？很高兴的告诉你，能！

Context有个createPackageContext方法，可以创建另外一个包的上下文，这个实例不同于它本身的Context实例，但是功能是一样的。

这个方法有两个参数：

1. packageName 包名，要得到Context的包名

2. flags 标志位，有CONTEXT_INCLUDE_CODE和CONTEXT_IGNORE_SECURITY两个选项。

CONTEXT_INCLUDE_CODE的意思是包括代码，也就是说可以执行这个包里面的代码。

CONTEXT_IGNORE_SECURITY的意思是忽略安全警告，如果不加这个标志的话，有些功能是用不了的，会出现安全警告。

下面给个例子，执行另外一个包里面的某个类的方法。

另外一个包的包名是com.iaiai，类名Main，方法名print，代码如下：

```
package com.iaiai;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;

class Main extends Activity {

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

    public void print(String msg) {
        Log.d("Main", "msg:" + msg);
    }
}
```

本包的调用Main的print方法的代码块如下：

```
try {
    Context c = createPackageContext("com.iaiai",
        Context.CONTEXT_INCLUDE_CODE
        | Context.CONTEXT_IGNORE_SECURITY);

    // 载入这个类
    Class clazz = c.getClassLoader().loadClass("com.iaiai.Main");
    // 新建一个实例
    Object owner = clazz.newInstance();
    // 获取print方法，传入参数并执行
    Object obj = clazz.getMethod("print", String.class).invoke(owner,
        "Hello");
} catch (Exception e) {
}
```

ok，这样，我们就调用了com.iaiai包的Main类的print方法，执行结果，打印出了Hello。

怎么样，这只是一个调用其他包的代码的例子，我们获取到Context，还可以做很多事情，当然，题目所说的坏事，还是不要做为好。

1.76 Android有效解决加载大图片时内存溢出的问题

发表时间: 2011-08-04

尽量不要使用setImageBitmap或setImageResource或BitmapFactory.decodeResource来设置一张大图，因为这些函数在完成decode后，最终都是通过java层的createBitmap来完成的，需要消耗更多内存。

因此，改用先通过BitmapFactory.decodeStream方法，创建出一个bitmap，再将其设为ImageView的source，

decodeStream最大的秘密在于其直接调用JNI>>nativeDecodeAsset()来完成decode，无需再使用java层的createBitmap，从而节省了java层的空间。

如果在读取时加上图片的Config参数，可以有效减少加载的内存，从而有效阻止抛出out of Memory异常

另外，decodeStream直接拿的图片来读取字节码了，不会根据机器的各种分辨率来自动适应，

使用了decodeStream之后，需要在hdpi和mdpi，ldpi中配置相应的图片资源，

否则在不同分辨率机器上都是同样大小（像素点数量），显示出来的大小就不对了。

另外，以下方式也大有帮助：

1.

```
InputStream is = this.getResources().openRawResource(R.drawable.pic1);
BitmapFactory.Options options=new BitmapFactory.Options();
options.inJustDecodeBounds = false;
options.inSampleSize = 10;    //width, height设为原来的十分之一
Bitmap btp =BitmapFactory.decodeStream(is,null,options);
```

2.

```
if(!bmp.isRecycle() ){
    bmp.recycle()    //回收图片所占的内存
    system.gc()    //提醒系统及时回收
}
```

以下奉上一个方法：

```
/**
 * 以最省内存的方式读取本地资源的图片
```

```
* @param context
* @param resId
* @return
*/
public static Bitmap readBitMap(Context context, int resId){
    BitmapFactory.Options opt = new BitmapFactory.Options();
    opt.inPreferredConfig = Bitmap.Config.RGB_565;
    opt.inPurgeable = true;
    opt.inInputShareable = true;
    //获取资源图片
    InputStream is = context.getResources().openRawResource(resId);
    return BitmapFactory.decodeStream(is,null,opt);
}
```

Android内存溢出的解决办法

转自：<http://www.cppblog.com/iuranus/archive/2010/11/15/124394.html?opt=admin>

昨天在模拟器上给gallery放入图片的时候，出现java.lang.OutOfMemoryError: bitmap size exceeds VM budget 异常，图像大小超过了RAM内存。

模拟器RAM比较小，只有8M内存，当我放入的大量的图片（每个100多K左右），就出现上面的原因。由于每张图片先前是压缩的情况，放入到Bitmap的时候，大小会变大，导致超出RAM内存，具体解决办法如下：

//解决加载图片 内存溢出的问题

```
//Options 只保存图片尺寸大小，不保存图片到内存
BitmapFactory.Options opts = new BitmapFactory.Options();
//缩放的比例，缩放是很难按准备的比例进行缩放的，其值表明缩放的倍数，SDK中建议其值是2的
//指数值,值越大会导致图片不清晰
opts.inSampleSize = 4;
Bitmap bmp = null;
bmp = BitmapFactory.decodeResource(getResources(),
mImageIds[position],opts);
```

...

```
//回收  
bmp.recycle();
```

通过上面的方式解决了，但是这并不是最完美的解决方式。

通过一些了解，得知如下：

优化Dalvik虚拟机的堆内存分配

对于Android平台来说，其托管层使用的Dalvik Java VM从目前的表现来看还有很多地方可以优化处理，比如我们在开发一些大型游戏或耗资源的应用中可能考虑手动干涉GC处理，使用 `dalvik.system.VMRuntime`类提供的`setTargetHeapUtilization`方法可以增强程序堆内存的处理效率。当然具体原理我们可以参考开源工程，这里我们仅说下使用方法：`private final static float TARGET_HEAP_UTILIZATION = 0.75f;` 在程序`onCreate`时就可以调用 `VMRuntime.getRuntime().setTargetHeapUtilization(TARGET_HEAP_UTILIZATION);` 即可。

Android堆内存也可自己定义大小

对于一些Android项目，影响性能瓶颈的主要是Android自己内存管理机制问题，目前手机厂商对RAM都比较吝啬，对于软件的流畅性来说RAM对性能的影响十分敏感，除了优化Dalvik虚拟机的堆内存分配外，我们还可以强制定义自己软件的对内存大小，我们使用Dalvik提供的 `dalvik.system.VMRuntime`类来设置最小堆内存为例：

```
private final static int CWJ_HEAP_SIZE = 6* 1024* 1024 ;
```

`VMRuntime.getRuntime().setMinimumHeapSize(CWJ_HEAP_SIZE);` //设置最小heap内存为6MB大小。当然对于内存吃紧来说还可以通过手动干涉GC去处理

bitmap 设置图片尺寸，避免内存溢出 `OutOfMemoryError`的优化方法

★android 中用bitmap 时很容易内存溢出，报如下错误：`Java.lang.OutOfMemoryError : bitmap size exceeds VM budget`

- 主要是加上这段：

```
BitmapFactory.Options options = new BitmapFactory.Options();  
options.inSampleSize = 2;
```

- eg1 : (通过Uri取图片)

```
private ImageView preview;
BitmapFactory.Options options = new BitmapFactory.Options();
    options.inSampleSize = 2;//图片宽高都为原来的二分之一，即图片为原来的四分之一
    Bitmap bitmap = BitmapFactory.decodeStream(cr
        .openInputStream(uri), null, options);
    preview.setImageBitmap(bitmap);
```

以上代码可以优化内存溢出，但它只是改变图片大小，并不能彻底解决内存溢出。

- eg2: (通过路径去图片)

```
private ImageView preview;
private String fileName= "/sdcard/DCIM/Camera/2010-05-14 16.01.44.jpg";
BitmapFactory.Options options = new BitmapFactory.Options();
    options.inSampleSize = 2;//图片宽高都为原来的二分之一，即图片为原来的四分之一
    Bitmap b = BitmapFactory.decodeFile(fileName, options);
    preview.setImageBitmap(b);
    filePath.setText(fileName);
```

★Android 还有一些性能优化的方法：

- 首先内存方面，可以参考 Android堆内存也可自己定义大小 和 优化Dalvik虚拟机的堆内存分配
- 基础类型上，因为Java没有实际的指针，在敏感运算方面还是要借助NDK来完成。Android123提示游戏开发者，这点比较有意思的是Google 推出NDK可能是帮助游戏开发人员，比如OpenGL ES的支持有明显的改观，本地代码操作图形界面是很必要的。
- 图形对象优化，这里要说的是Android上的Bitmap对象销毁，可以借助recycle()方法显示让GC回收一个Bitmap对象，通常对一个不用的Bitmap可以使用下面的方式，如

```
if(bitmapObject.isRecycled() == false) //如果没有回收
    bitmapObject.recycle();
```

- 目前系统对动画支持比较弱智对于常规应用的补间过渡效果可以，但是对于游戏而言一般的美工可能习惯了GIF方式的统一处理，目前Android系统仅能预览GIF的第一帧，可以借助J2ME中通过线程和自己写解析器的方式来读取GIF89格式的资源。
- 对于大多数Android手机没有过多的物理按键可能我们需要想象下了做好手势识别 GestureDetector 和重力感应来实现操控。通常我们还要考虑误操作问题的降噪处理。

Android堆内存也可自己定义大小

对于一些大型Android项目或游戏来说在算法处理上没有问题外，影响性能瓶颈的主要是Android自己内存管理机制问题，目前手机厂商对RAM都比较吝啬，对于软件的流畅性来说RAM对性能的影响十分敏感，除了上次Android开发网提到的 优化Dalvik虚拟机的堆内存分配外，我们还可以强制定义自己软件的对内存大小，我们使用Dalvik提供的 `dalvik.system.VMRuntime`类来设置最小堆内存为例：

```
private final static int CWJ_HEAP_SIZE = 6* 1024* 1024 ;
```

`VMRuntime.getRuntime().setMinimumHeapSize(CWJ_HEAP_SIZE);` //设置最小heap内存为6MB大小。当然对于内存吃紧来说还可以通过手动干涉GC去处理，我们将在下次提到具体应用。

优化Dalvik虚拟机的堆内存分配

对于Android平台来说，其托管层使用的Dalvik JavaVM从目前的表现来看还有很多地方可以优化处理，比如我们在开发一些大型游戏或耗资源的应用中可能考虑手动干涉GC处理，使用 `dalvik.system.VMRuntime`类提供的`setTargetHeapUtilization`方法可以增强程序堆内存的处理效率。当然具体原理我们可以参考开源工程，这里我们仅说下使用方法：`private final static float TARGET_HEAP_UTILIZATION = 0.75f;` 在程序`onCreate`时就可以调用 `VMRuntime.getRuntime().setTargetHeapUtilization(TARGET_HEAP_UTILIZATION);` 即可。

介绍一下图片占用进程的内存算法吧。

android中处理图片的基础类是Bitmap，顾名思义，就是位图。占用内存的算法如下：

图片的`width*height*Config`。

如果`Config`设置为`ARGB_8888`，那么上面的`Config`就是4。一张`480*320`的图片占用的内存就是`480*320*4 byte`。

前面有人说了一下8M的概念，其实是在默认情况下android进程的内存占用量为16M，因为Bitmap他除了java中持有数据外，底层C++的 skia图形库还会持有一个SKBitmap对象，因此一般图片占用内存推荐大小应该不超过8M。这个可以调整，编译源代码时可以设置参数。

1.77 Android善用预定义样式

发表时间: 2011-08-05

字体大小

对于能够显示文字的控件（如TextView EditText RadioButton Button CheckBox Chronometer等等），你有时需要控制字体的大小。Android平台定义了三种字体大小。

```
"?android:attr/textAppearanceLarge"
```

```
"?android:attr/textAppearanceMedium"
```

```
"?android:attr/textAppearanceSmall"
```

使用方法为：

```
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"
```

字体颜色

```
android:textColor="?android:attr/textColorPrimary"
```

```
android:textColor="?android:attr/textColorSecondary"
```

```
android:textColor="?android:attr/textColorTertiary"
```

```
android:textColor="?android:attr/textColorPrimaryInverse"
```

```
android:textColor="?android:attr/textColorSecondaryInverse"
```

ProgressBar

```
style="?android:attr/progressBarStyleHorizontal"
```

```
style="?android:attr/progressBarStyleLarge"
```

```
style="?android:attr/progressBarStyleSmall"  
style="?android:attr/progressBarStyleSmallTitle"
```

分隔符

横向：

```
<View android:layout_width="fill_parent"  
      android:layout_height="1dip"  
      android:background="?android:attr/listDivider" />
```

纵向：

```
<View android:layout_width="1dip"  
      android:layout_height="fill_parent"  
      android:background="?android:attr/listDivider" />
```

CheckBox

```
style="?android:attr/starStyle"
```

类似标题栏效果的TextView

```
style="?android:attr/listSeparatorTextViewStyle"
```

其它有用的样式

```
android:layout_height="?android:attr/listPreferredItemHeight"
```

```
android:paddingRight="?android:attr/scrollbarSize"
```

```
style="?android:attr/windowTitleBackgroundStyle"
```

```
style="?android:attr/windowTitleStyle"
```

```
android:layout_height="?android:attr/windowTitleSize"
```

```
android:background="?android:attr/windowBackground"
```

1.78 Android高手应该精通哪些内容?

发表时间: 2011-08-05

很多Android开发者已经度过了初级、中级，如何成为一个Android高手呢？Android123就各个级别的程序员应该掌握哪些内容作为下面分类。

一、初级

1. 拥有娴熟的Java基础，理解设计模式，比如OOP语言的工厂模式要懂得。
2. 掌握Android UI控件、Android Java层API相关使用。

迈向中级，最好再次更新下Java技术，掌握Java concurrent库、nio库，精通xml、json解析，做过socket c/s端的程序。

二、中级

1. 熟练掌握NDK JNI方式的开发，对Android framework十分了解。
2. 移植一些开源项目，比如FFmpeg、Mplayer到Android上运行。

迈向高级，熟悉android开源代码的结构，中间件下层的库，比如webkit、opencore、opengl。

三、高级

1. 熟练的Linux驱动开发，精通Linux内核结构。
2. 掌握Android移植，比如常见的TI OMAP、Qualcomm MSM的硬件移植。

迈向顶级，发现Android系统中的0day漏洞，可以轻松的找出Root系统的10种方法，帮助Google改进Android操作系统。

1.79 Activity 间传递对象, 采用Parcelable

发表时间: 2011-08-06

android提供了一种新的类型 : Parcel。本类被用作封装数据的容器, 封装后的数据可以通过Intent或IPC传递。除了基本类型以外, 只有实现了Parcelable接口的类才能被放入Parcel中。

Parcelable实现要点 : 需要实现三个东西

- writeToParcel 方法。该方法将类的数据写入外部提供的Parcel中. 声明如下 :

引用

writeToParcel (Parcel dest, int flags) 具体参数含义见javadoc

- describeContents方法。没搞懂有什么用, 反正直接返回0也可以
- 静态的Parcelable.Creator接口, 本接口有两个方法 :

引用

createFromParcel(Parcel in) 实现从in中创建出类的实例的功能

newArray(int size) 创建一个类型为T, 长度为size的数组, 仅一句话 (return new T[size])即可。估计本方法是供外部类反序列化本类数组使用。

测试用的接收信息Activity

```
import com.iaiai;

import android.content.Intent;

import android.os.Bundle;
import android.os.Parcelable;

/** 
 * 
 * <p>
 * Title: Test.java
 * </p>
 * <p>
```

```
* E-Mail: 176291935@qq.com
* </p>
* <p>
* QQ: 176291935
* </p>
* <p>
* Http: iaiai.iteye.com
* </p>
* <p>
* Create time: 2011-7-21
* </p>
*
* @author 丸子
* @version 0.0.1
*/
public class Test extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Intent i = getIntent();
        Person p = i.getParcelableExtra("yes");
        System.out.println("---->" + p.name);
        System.out.println("---->" + p.map.size());
    }
}
```

发送的Activity

```
import com.iaiai;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
```

```
/**  
 *  
 * <p>  
 * Title: TestNew.java  
 * </p>  
 * <p>  
 * E-Mail: 176291935@qq.com  
 * </p>  
 * <p>  
 * QQ: 176291935  
 * </p>  
 * <p>  
 * Http: iaiai.iteye.com  
 * </p>  
 * <p>  
 * Create time: 2011-7-21  
 * </p>  
 *  
 * @author 丸子  
 * @version 0.0.1  
 */  
  
public class TestNew extends Activity {  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
        Intent intent = new Intent();  
        Person p = new Person();  
        p.map = new HashMap<String, String>();  
        p.map.put("yes", "ido");  
        p.name="ok";  
        intent.putExtra("yes", p);  
        intent.setClass(this, Test.class);  
        startActivity(intent);  
    }  
}
```

```
    }  
}
```

Parcelable的实现类

```
import java.util.HashMap;  
  
import android.os.Parcel;  
import android.os.Parcelable;  
  
/**  
 *  
 * <p>  
 * Title: Person.java  
 * </p>  
 * <p>  
 * E-Mail: 176291935@qq.com  
 * </p>  
 * <p>  
 * QQ: 176291935  
 * </p>  
 * <p>  
 * Http: iaiai.iteye.com  
 * </p>  
 * <p>  
 * Create time: 2011-7-21  
 * </p>  
 *  
 * @author 丸子  
 * @version 0.0.1  
 */  
  
public class Person implements Parcelable {  
  
    public HashMap<String, String> map = new HashMap<String, String>();
```

```
public String name ;  
@Override  
public int describeContents() {  
    return 0;  
}  
  
@Override  
public void writeToParcel(Parcel dest, int flags) {  
  
    dest.writeMap(map);  
    dest.writeString(name);  
}  
public static final Parcelable.Creator<Person> CREATOR = new Parcelable.Creator<Person>() {  
    //重写Creator  
  
    @Override  
    public Person createFromParcel(Parcel source) {  
        Person p = new Person();  
        p.map=source.readHashMap(HashMap.class.getClassLoader());  
        p.name=source.readString();  
        return p;  
    }  
  
    @Override  
    public Person[] newArray(int size) {  
        // TODO Auto-generated method stub  
        return null;  
    }  
};  
}
```

1.80 Android NDK简介

发表时间: 2011-08-06

Android NDK是一个工具集，让你的Android应用程序里可以内嵌使用本地代码（C/C++）的组件。Android应用程序运行在Dalvik虚拟机中。NDK可以让你使用C/C++这样的本地代码语言来实现你的应用程序中某些部分。这对某类程序是有帮助的，比如需要重用已有的C代码，或者为了提高运行速度。

NDK 提供:

编译文件和工具集，用来将你的C/C++源文件编译成本地库。

提供一种方式，将对应的本地库内嵌到应用程序包文件（.apk）中，最终发布到Android设备中。

本地系统头文件和库，这些头文件和库从Android 1.5开始往后都是被支持的。但使用本地活动(native activity)的程序只能运行在Android 2.3或更高的系统中。

文档、示例、指南。

最近发行的NDK版本支持以下的ARM指令集:

ARMv5TE (包括 Thumb-1 指令)

ARMv7-A (包括 Thumb-2 和 VFPv3-D16 指令, 可选地支持 NEON/VFPv3-D32)

后续的NDK将支持:

x86 指令 (查阅 CPU-ARCH-ABIS.HTML获得更多信息)

1.81 Android 总结4种线程中操作UI界面的方法

发表时间: 2011-08-07

我们经常会在后台线程中去做一些耗时的操作，比如去网络取数据。但是当数据取回来，需要显示到页面上的时候，会遇到一些小麻烦，因为我们都知道，android的UI页面是不允许在其他线程直接操作的。下面总结4中方法用来在线程中操作UI界面。

方法一：Handler

子线程调用Handler的sendMessage(message)发送事件。

```
mHandler = new Handler() {  
    @Override  
    public void handleMessage(Message msg) {  
        //操作界面  
        myText.setText( 来自网络的信息 );  
        super.handleMessage(msg);  
    }  
};
```

```
public class MyThread extends Thread {  
    public void run() {  
        // 耗时操作  
        loadNetwork();  
        Message msg = new Message();  
        mHandler.sendMessage(msg); //向Handler发送消息 ,  
    }  
}
```

方法二：View.post (Runnable)

```
public void onClick( View v ) {  
    new Thread( new Runnable() {
```

```
public void run() {  
    // 耗时操作  
    loadNetwork();  
    myText.( new Runnable() {  
        myText.setText( 来自网络的信息 );  
    });  
}  
}).start();  
}
```

方法三 : Activity.runOnUiThread (Runnable)

与view.post类似

```
public void onClick( View v ) {  
    new Thread( new Runnable() {  
        public void run() {  
            // 耗时操作  
            loadNetwork();  
            Activity.runOnUiThread( new Runnable() {  
                myText.setText( 来自网络的信息 );  
            });  
        }  
    }).start();  
}
```

方法四 : AsyncTask

```
//UI线程中执行  
new DownloadImageTask().execute( "www.91dota.com" );
```

```
private class LoadTask extends AsyncTask {  
    protected String doInBackground( String... url ) {  
        return loadDataFromNetwork( url[0] );//后台耗时操作  
    }  
  
    protected void onPostExecute( String result ) {  
        myText.setText( result ); //得到来自网络的信息刷新页面  
    }  
}
```

1.82 Android获取状态栏和标题栏的高度

发表时间: 2011-08-08

1.获取状态栏高度：

decorView是window中的最顶层view，可以从window中获取到decorView，然后decorView有个getWindowVisibleDisplayFrame方法可以获取到程序显示的区域，包括标题栏，但不包括状态栏。于是，我们就可以算出状态栏的高度了。

```
Rect frame = new Rect();
getWindow().getDecorView().getWindowVisibleDisplayFrame(frame);
int statusBarHeight = frame.top;
```

```
Rect frame = new Rect();
getWindow().getDecorView().getWindowVisibleDisplayFrame(frame);
int statusBarHeight = frame.top;
```

2.获取标题栏高度：

getWindow().findViewById(Window.ID_ANDROID_CONTENT)这个方法获取到的view就是程序不包括标题栏的部分，然后就可以知道标题栏的高度了。

```
int contentTop = getWindow().findViewById(Window.ID_ANDROID_CONTENT).getTop();
//statusBarHeight是上面所求的状态栏的高度
int titleBarHeight = contentTop - statusBarHeight
```

```
int contentTop = getWindow().findViewById(Window.ID_ANDROID_CONTENT).getTop();
//statusBarHeight是上面所求的状态栏的高度
int titleBarHeight = contentTop - statusBarHeight
```

1.83 Android 获取手机屏幕大小两种方法

发表时间: 2011-08-08

第一种：

```
public Point getDisplayMetrics() {  
    DisplayMetrics dm = new DisplayMetrics();  
    dm = getApplicationContext().getResources().getDisplayMetrics();  
    int screenWidth = dm.widthPixels;  
    int screenHeight = dm.heightPixels;  
    return new Point(screenWidth, screenHeight);  
}
```

第二种：

```
public Point getDisplayScreen() {  
    WindowManager windowManager = getWindowManager();  
    Display display = windowManager.getDefaultDisplay();  
    screenWidth = display.getWidth();  
    screenHeight = display.getHeight();  
    return new Point(screenWidth, screenHeight);  
}
```

1.84 android:clearTaskOnLaunch="true"什么意思？

发表时间: 2011-08-08

```
android:clearTaskOnLaunch="true"
```

这个参数意思是无论何时这个activity都会从home screen重新启动，也就是只要你按了桌面的图标，这个活动就是第一个发起的。但是唯一要注意的是如果你按下的是home键那么意味着这个活动没有结束，所以你在按有时候就不会第一次发起了。只有当你按下back或者结束当前执行的任务之类代表能使activity结束的行为这个活动都会第一次发起。

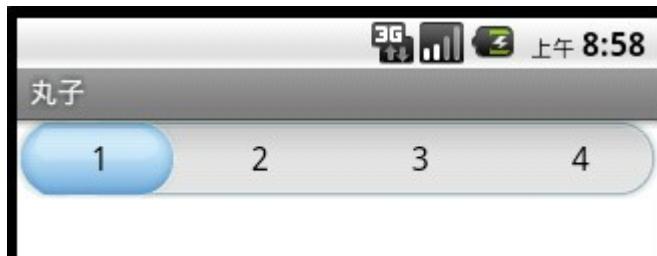
往往 android:clearTaskOnLaunch="true" 是一个主入口。

```
<activity android:name="Camera"
    android:theme="@android:style/Theme.Black.NoTitleBar.Fullscreen"
    android:clearTaskOnLaunch="true"
    android:taskAffinity="android.task.camera">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.media.action.IMAGE_CAPTURE" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.media.action.STILL_IMAGE_CAMERA" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

1.85 Android TabSwitcher自定义控件

发表时间: 2011-08-11

先来看下效果:



main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent" android:background="#ffffffff">
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/tab" android:orientation="vertical"
        android:layout_width="fill_parent" android:layout_height="wrap_content"
        android:background="#ffffffff">
    </LinearLayout>
</LinearLayout>
```

TabSwitcher.java

```
package com.iaiai.activity;

import android.content.Context;
import android.graphics.Color;
import android.view.Gravity;
import android.view.View;
import android.widget.LinearLayout;
import android.widget.TextView;

/**
 *
```

```
* <p>
* Title: TabSwitcher.java
* </p>
* <p>
* E-Mail: 176291935@qq.com
* </p>
* <p>
* QQ: 176291935
* </p>
* <p>
* Http: iaiai.iteye.com
* </p>
* <p>
* Create time: 2011-8-11
* </p>
*
* @author 丸子
* @version 0.0.1
*/
public class TabSwitcher extends LinearLayout {

    private Context context;
    private String[] texts;
    private int selectedPosition = 0;
    private int oldPosition = selectedPosition;
    private int[] background = { R.drawable.tabswitcher_short,
            Color.TRANSPARENT };
    private TextView[] tvs;

    public TabSwitcher(Context context, String[] texts) {
        super(context);
        this.texts = texts;
        this.context = context;
        setOrientation(HORIZONTAL);
        LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.FILL_PARENT,
                LinearLayout.LayoutParams.WRAP_CONTENT);
    }
}
```

```
        setLayoutParams(params);
        setBackgroundResource(R.drawable.tabswitcher_long);
    }

@Override
protected void onFinishInflate() {
    // TODO Auto-generated method stub
    super.onFinishInflate();
    tvs = new TextView[texts.length];
    LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(
        LinearLayout.LayoutParams.FILL_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT);
    params.weight = 1;
    params.gravity = Gravity.CENTER_VERTICAL;
    for (int i = 0; i < texts.length; i++) {
        TextView child = new TextView(context);
        child.setTag(i);
        child.setTextSize(16);
        child.setTextColor(Color.BLACK);
        child.setGravity(Gravity.CENTER);

        tvs[i] = child;
        child.setOnClickListener(listener);
        this.addView(child, params);
    }
}

OnClickListener listener = new OnClickListener() {

    @Override
    public void onClick(View v) {
        selectedPosition = (Integer) v.getTag();
        if (selectedPosition != oldPosition) {
            tvs[oldPosition].setBackgroundColor(background[1]);
            oldPosition = selectedPosition;
            ((TextView) v).setBackgroundColor(background[0]);
            if (onItemClickLisener != null) {

```

```
        onItemClickListener.onItemClickListener(v, selectedPosition);
    }
}

};

@Override
protected void onSizeChanged(int w, int h, int oldw, int oldh) {
    super.onSizeChanged(w, h, oldw, oldh);
    oldPosition = selectedPosition;
    for (int i = 0; i < texts.length; i++) {
        tvs[i].setText(texts[i]);
        if (selectedPosition == i) {
            tvs[i].setBackgroundResource(background[0]);
        } else {
            tvs[i].setBackgroundColor(background[1]);
        }
    }
}

private OnItemClickListener onItemClickListener;

public void setOnItemClickListener(OnItemClickListener onItemClickListener) {
    this.onItemClickListener = onItemClickListener;
}

public interface OnItemClickListener {
    void onItemClickListener(View view, int position);
}

public void setTexts(String[] texts) {
    this.texts = texts;
}

public void setSelectedPosition(int selectedPosition) {
    this.selectedPosition = selectedPosition;
}
```

}

}

IaiaiActivity.java

```
package com.iaiai.activity;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.LinearLayout;
import android.widget.Toast;

import com.iaiai.activity.TabSwitcher.OnItemClickLisener;

/**
 *
 * <p>
 * Title: IaiaiActivity.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-8-11
 * </p>
 *
 * @author 丸子
```

```
* @version 0.0.1
*/
public class IaiaiActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        TabSwitcher tabSwitcher = new TabSwitcher(this, new String[] { "1",
                "2", "3", "4" });
        tabSwitcher.setSelectedPosition(1);
        tabSwitcher.setOnItemClickLisener(onItemClickLisener);
        tabSwitcher.onFinish Inflate();

        LinearLayout linearLayout = (LinearLayout) findViewById(R.id.tab);
        linearLayout.addView(tabSwitcher, new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.FILL_PARENT,
                LinearLayout.LayoutParams.WRAP_CONTENT));
    }

    OnItemClickLisener onItemClickLisener = new OnItemClickLisener() {
        @Override
        public void onItemClickLisener(View view, int position) {
            // TODO Auto-generated method stub
            //
            switch (position) {
            case 0:
            case 1:
            case 2:
                Toast.makeText(IaiaiActivity.this,
                        "position clicked:" + position, Toast.LENGTH_SHORT
                        .show());
                break;
            default:
                break;
            }
        }
    }
}
```

```
};  
}
```

tabswitcher_long.9.png:



tabswitcher_short.9.png:



1.86 Android 左右滑动 控件

发表时间: 2011-08-12

先看效果:



main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView android:id="@+id/textView" android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:text="ddd" />
    <com.diydyq.android.swipeTest.SlipView
        android:id="@+id/slipView" android:orientation="vertical"
        android:layout_width="fill_parent" android:layout_height="fill_parent">
    </com.diydyq.android.swipeTest.SlipView>
</LinearLayout>
```

SlipEntity.java

```
package com.iaiai.activity;

import java.util.HashMap;
```

```
import java.util.Map;

import android.graphics.Bitmap;
import android.util.Log;

/***
 *
 * <p>
 * Title: SlipEntity.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-8-11
 * </p>
 *
 * @author 丸子
 * @version 0.0.1
 */
public class SlipEntity {

    public static final int FLAG_BACK = 1;
    public static final int FLAG_FROT = 2;
    public static final float DOCK_L = 0;
    public static final float DOCK_M = (float) 0.5;
    public static final float DOCK_R = 1;
    public static final float MICRO_X = 10;
    /** Background image */

    Bitmap backImage;
    /** Front image */
```

```
Bitmap frotImage;

/** Start Position of back image */
float xBackStart;

/** Start Position of back image */
float yBackStart;

/** Start Position of front image */
float xFrotStart;

/** Start Position of front image */
float yFrotStart;

/** initial Position of front image */
float xFrotInitPos;

/** initial Position of front image */
float yFrotInitPos;

/** Margin of front and back image in X-Axis */
float xMarginLeft;

/** Margin of front and back image in Y-Axis */
float yMarginTop;

/** Containing dock position of the front image */

Map<Float, Float> dockPosList = new HashMap<Float, Float>();

/** Current dock Percentage: DOCK_L | DOCK_M | DOCK_R */
float curDockPer = DOCK_L;

/** Weather has invoked initSlipEntity() */
boolean isInit = false;

public SlipEntity() {

}

public SlipEntity(Bitmap backImage, Bitmap frotImage) {
    this.backImage = backImage;
    this.frotImage = frotImage;
}

public SlipEntity(float xBackStart, float yBackStart, float xFrotStart,
                 float yFrotStart) {
    this.xBackStart = xBackStart;
    this.yBackStart = yBackStart;
```

```
        this.xFrotStart = xFrotStart;
        this.yFrotStart = yFrotStart;
    }

    public void initSlipEntity(float viewWidth, float viewHeight) {
        this.xBackStart = (viewWidth - this.backImage.getWidth()) / 2;
        this.yBackStart = (viewHeight - this.backImage.getHeight()) / 2;
        this.xMarginLeft = 5;
        this.yMarginTop = (this.backImage.getHeight() - this.frotImage
                .getHeight()) / 2;
        this.xFrotInitPos = this.xBackStart + this.xMarginLeft;
        this.yFrotInitPos = this.yBackStart + this.yMarginTop;
        this.xFrotStart = this.xFrotInitPos;
        this.yFrotStart = this.yFrotInitPos;
        // Add dock position
        float dockL = this.xFrotInitPos;
        float dockR = this.xBackStart + this.backImage.getWidth()
                - this.frotImage.getWidth() - this.xMarginLeft;
        this.dockPosList.put(DOCK_L, dockL);
        this.dockPosList.put(DOCK_R, dockR);
        for (Float dockPer : this.dockPosList.keySet()) {
            if (this.dockPosList.get(dockPer) == 0) {
                float docPos = (dockR - dockL) * dockPer
                        + this.dockPosList.get(DOCK_L);
                this.dockPosList.put(dockPer, docPos);
            }
        }
    }

    // Dock at current position
    this.xFrotStart = this.dockPosList.get(this.curDockPer);
    this.isInit = true;
    // For debug information
    StringBuilder sb = new StringBuilder();
    sb.append("BackImageW:" + this.backImage.getWidth() + "\n");
    sb.append("BackImageH:" + this.backImage.getHeight() + "\n");
    sb.append("FrotImageW:" + this.frotImage.getWidth() + "\n");
    sb.append("FrotImageH:" + this.frotImage.getHeight() + "\n");
}
```

```
        sb.append("xBackStart:" + xBackStart + "\n");
        sb.append("yBackStart:" + yBackStart + "\n");
        sb.append("xMarginLeft:" + xMarginLeft + "\n");
        sb.append("yMarginTop:" + yMarginTop + "\n");
        sb.append("xFrotInitP:" + xFrotInitPos + "\n");
        sb.append("yFrotInitP:" + yFrotInitPos + "\n");
        sb.append("xFrotStart:" + xFrotStart + "\n");
        sb.append("yFrotStart:" + yFrotStart + "\n");
        Log.v("SlipEntity", sb.toString()));

    }

    /**
     * Weather the front image reaches the max right of background image, if
     * true, set xFrotStart to max right.
     *
     * @return
     */
    public boolean isReachRight() {
        if (this.xFrotStart > this.dockPosList.get(DOCK_R)) {
            this.curDockPer = DOCK_R;
            this.xFrotStart = this.dockPosList.get(DOCK_R);
            return true;
        } else {
            return false;
        }
    }

    /**
     * Weather the front image reaches the max left of background image, if
     * true, set xFrotStart to max left.
     *
     * @return
     */
    public boolean isReachLeft() {
        if (this.xFrotStart < this.dockPosList.get(DOCK_L)) {
            this.curDockPer = DOCK_L;
            this.xFrotStart = this.dockPosList.get(DOCK_L);
        }
    }
}
```

```
        return true;
    } else {
        return false;
    }
}

/**
 * Weather the point(x,y) is in the area of back or front image
 *
 * @param type
 *          FLAG_FROT(front image) | FLAG_BACK(back image)
 * @param x
 *          X-coordinate of point
 * @param y
 *          Y-coordinate of point
 * @return weather the point is in specified area
 */
public boolean isPointInImage(int type, float x, float y) {
    float rPointX;
    float rPointY;
    switch (type) {
        case FLAG_FROT:
            rPointX = this.xFrotStart + this.frotImage.getWidth();
            rPointY = this.yFrotStart + this.frotImage.getHeight();
            if (x > this.xFrotStart && y > this.yFrotStart && x < rPointX
                && y < rPointY)
                return true;
            else
                return false;
        case FLAG_BACK:
            rPointX = this.xBackStart + this.backImage.getWidth();
            rPointY = this.yBackStart + this.backImage.getHeight();
            if (x > this.xBackStart && y > this.yBackStart && x < rPointX
                && y < rPointY)
                return true;
            else
                return false;
    }
}
```

```
        default:
            return false;
        }
    }

    /**
     * Is the current touch in some dock position
     *
     * @return return dockPer if in, or -1 while no match
     */
    public float isDock() {
        for (float dockPer : this.dockPosList.keySet()) {
            float dockPos = this.dockPosList.get(dockPer);
            if (this.xFrotStart > dockPos - MICRO_X
                && this.xFrotStart < dockPos + MICRO_X) {
                this.curDockPer = dockPer;
                return dockPer;
            }
        }
        return -1;
    }

    /**
     * Get the current dock percentage in x-axis
     *
     * @return
     */
    public float getCurDockPer() {
        return this.curDockPer;
    }

    /**
     * Get the current dock position in x-axis
     *
     * @return
     */
    public float getCurDockPos() {
```

```
        return this.dockPosList.get(this.curDockPer);  
    }  
  
    /**  
     * Add dock position to the list  
     *  
     * @param dockPer  
     *          dock Percent should be between (0.0,1.0)  
     */  
  
    public void addDockPos(float dockPer) {  
        if (dockPer > 0 && dockPer < 1) {  
            this.dockPosList.put(dockPer, (float) 0.0);  
        }  
    }  
  
    /**  
     * Return width of background image  
     *  
     * @return  
     */  
  
    public float getBackWidth() {  
        return this.backImage.getWidth();  
    }  
  
    /**  
     * Return height of background image  
     *  
     * @return  
     */  
    public float getBackHeight() {  
        return this.backImage.getHeight();  
    }  
  
    /**  
     * Return width of front image
```

```
*  
* @return  
*/  
  
public float getFrontWidth() {  
    return this.frontImage.getWidth();  
}  
  
/**  
 * Return height of front image  
 *  
 * @return  
*/  
  
public float getFrontHeight() {  
    return this.frontImage.getWidth();  
}  
  
/**  
 * Dock at some position  
 *  
 * @param curDockPer  
*/  
  
public void setCurDockPos(float curDockPer) {  
    this.curDockPer = curDockPer;  
}  
  
public Bitmap getBackImage() {  
    return backImage;  
}  
  
public void setBackImage(Bitmap backImage) {  
    this.backImage = backImage;  
}  
  
public Bitmap getFrontImage() {  
    return frontImage;  
}
```

```
public void setFrontImage(Bitmap frontImage) {  
    this.frontImage = frontImage;  
}  
  
public float getxBackStart() {  
    return xBackStart;  
}  
  
public void setxBackStart(float xBackStart) {  
    this.xBackStart = xBackStart;  
}  
  
public float getyBackStart() {  
    return yBackStart;  
}  
  
public void setyBackStart(float yBackStart) {  
    this.yBackStart = yBackStart;  
}  
  
public float getxFrontStart() {  
    return xFrontStart;  
}  
  
public void setxFrontStart(float xFrontStart) {  
    this.xFrontStart = xFrontStart;  
}  
  
public float getyFrontStart() {  
    return yFrontStart;  
}  
  
public void setyFrontStart(float yFrontStart) {  
    this.yFrontStart = yFrontStart;  
}
```

```
public float getxFrotInitPos() {
    return xFrotInitPos;
}

public void setxFrotInitPos(float xFrotInitPos) {
    this.xFrotInitPos = xFrotInitPos;
}

public float getyFrotInitPos() {
    return yFrotInitPos;
}

public void setyFrotInitPos(float yFrotInitPos) {
    this.yFrotInitPos = yFrotInitPos;
}

public float getxMarginLeft() {
    return xMarginLeft;
}

public void setxMarginLeft(float xMarginLeft) {
    this.xMarginLeft = xMarginLeft;
}

public float getyMarginTop() {
    return yMarginTop;
}

public void setyMarginTop(float yMarginTop) {
    this.yMarginTop = yMarginTop;
}

public Map<Float, Float> getDockPosList() {
    return dockPosList;
}

public void setDockPosList(Map<Float, Float> dockPosList) {
```

```
        this.dockPosList = dockPosList;
    }

    public boolean isInit() {
        return isInit;
    }

    public void setInit(boolean isInit) {

        this.isInit = isInit;
    }
}
```

SlipView.java

```
package com.iaiai.activity;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.drawable.BitmapDrawable;
import android.util.AttributeSet;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;

/**
 *
 * <p>
 * Title: SlipView.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>

```

```
* QQ: 176291935
* </p>
* <p>
* Http: iaiai.iteye.com
* </p>
* <p>
* Create time: 2011-8-11
* </p>
*
* @author 丸子
* @version 0.0.1
*/
public class SlipView extends View {
    private static final String TAG = "SlipView";
    /** Listen slip on the block, no matter the event is success or fail */
    private OnSlipListener onSlipListener;
    /** Slip entity to set the value about backImage, frontImage position */
    private SlipEntity slipEntity;
    private float tmpTouchX;
    private float tmpTouchGap;

    public SlipView(Context context) {
        super(context);
        Bitmap backImage = ((BitmapDrawable) getResources().getDrawable(
            R.drawable.back5)).getBitmap();
        Bitmap frontImage = ((BitmapDrawable) getResources().getDrawable(
            R.drawable.front1)).getBitmap();
        this.slipEntity = new SlipEntity(backImage, frontImage);
    }

    public SlipView(Context context, AttributeSet attr) {
        super(context, attr);
        Bitmap backImage = ((BitmapDrawable) getResources().getDrawable(
            R.drawable.back5)).getBitmap();
        Bitmap frontImage = ((BitmapDrawable) getResources().getDrawable(
            R.drawable.front1)).getBitmap();
        this.slipEntity = new SlipEntity(backImage, frontImage);
    }
}
```

```
}

public SlipView(Context context, SlipEntity slipEntity) {
    super(context);
    this.slipEntity = slipEntity;
}

@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    if (!this.slipEntity.isInit) {
        Log.v(TAG, "Init SlipEntity");
        this.slipEntity.initSlipEntity(this.getWidth(), this.getHeight());
    }
    canvas.drawBitmap(this.slipEntity.getBackImage(),
                      this.slipEntity.xBackStart, this.slipEntity.yBackStart, null);
    canvas.drawBitmap(this.slipEntity.getFrotImage(),
                      this.slipEntity.xFrotStart, this.slipEntity.yFrotStart, null);
}

/**
 * listen touch events and notify listener
 */

@Override
public boolean onTouchEvent(MotionEvent event) {

    Log.v(TAG, "Touch Position:" + event.getX());
    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN:
            // Log.v(TAG, "Down");
            if (this.slipEntity.isPointInImage(SlipEntity.FLAG_FROT,
                                              event.getX(), event.getY())) {
                this.tmpTouchX = event.getX();
                this.tmpTouchGap = event.getX() - this.slipEntity.xFrotStart;
            }
    }
}
```

```
        }

        break;

    case MotionEvent.ACTION_MOVE:
        // Log.v(TAG, "Move");
        this.slipEntity.isReachRight();
        this.slipEntity.isReachLeft();
        // if point(x,y) is not in back image, front image will not move
        if (this.slipEntity.isPointInImage(SlipEntity.FLAG_BACK,
                event.getX(), event.getY())) {
            // Log.v(TAG, "Move2");
            this.slipEntity.xFrotStart = event.getX() - this.tmpTouchGap;
        }
        break;

    case MotionEvent.ACTION_UP:
        Log.v(TAG, "Up");
        int flagLR = 0;
        if (event.getX() < this.tmpTouchX)
            flagLR = 1;
        else
            flagLR = 2;
        float dockPer = this.slipEntity.isDock();
        if (dockPer != -1) {
            // Dock at some Position
            if (this.onSlipListener != null)
                this.onSlipListener.slipDock(slipEntity, event,
                        this.slipEntity.getCurDockPer());
            if (event.getX() < this.tmpTouchX) {
                // Slip: <=
                if (dockPer == 0.0) {
                    // Reached
                    if (this.onSlipListener != null)
                        this.onSlipListener.slipLeft(slipEntity,
                                true);
                } else {
                    // Not Reached
                    this.slipEntity.xFrotStart = this.slipEntity
                            .getCurDockPos();
                }
            }
        }
    }
}
```

```
        if (this.onSlipListener != null)
            this.onSlipListener.slipLeft(slipEntity,
                false);
        }
    } else {
        // Slip: ==>
        if (dockPer == 1.0) {
            // Reached
            if (this.onSlipListener != null)
                this.onSlipListener.slipRight(slipEntity,
                    true);
        } else {
            // Not Reached
            this.slipEntity.xFrotStart = this.slipEntity
                .getCurDockPos();
            if (this.onSlipListener != null)
                this.onSlipListener.slipRight(slipEntity,
                    false);
        }
        break;
    }
} else {
    // No dock
    this.slipEntity.xFrotStart = this.slipEntity.getCurDockPos();
    if (flagLR == 1)
        this.onSlipListener.slipLeft(slipEntity, event, false);
    else
        this.onSlipListener.slipRight(slipEntity, event, false)
}
// if (event.getX() < this.tmpTouchX)
// {

    // // Slip: <= // if (this.slipEntity.isReachLeft())
    // { // // Reached

        // if (this.onSlipListener != null)
        // this.onSlipListener.slipLeft(slipEntity, event, true);
}
```

```
        // }

        // else

        // {

        // // Not Reached

        // this.slipEntity.xFrotStart = this.slipEntity.getCurDockPos();

        // if (this.onSlipListener != null)

        // this.onSlipListener.slipLeft(slipEntity, event, false);

        // }

        // }

        // else

        // {

        // // Slip: ==>

        // if (this.slipEntity.isReachRight())

        // {

        // // Reached

        // if (this.onSlipListener != null)

        // this.onSlipListener.slipRight(slipEntity, event, true);

        // }

        // else

        // {

        // // Not Reached

        // this.slipEntity.xFrotStart = this.slipEntity.getCurDockPos();

        // if (this.onSlipListener != null)

        // this.onSlipListener.slipRight(slipEntity, event, false);

        // }

        // break;

        // }

    }

    this.invalidate();

    return true;

}

/**

 * Listener on slip event on slippery view author diydyq

 *

 */

public interface OnSlipListener {
```

```
/**  
 * Listen a slip after touch down and up, not including touch move  
 *  
 * @param slipEntity  
 * @param event  
 * @param isSuccess  
 */  
  
public abstract void slipLeft(SlipEntity slipEntity, MotionEvent event,  
                                boolean isSuccess);  
  
/**  
 * Listen a slip after touch down and up, not including touch move  
 *  
 * @param slipEntity  
 * @param event  
 * @param isSuccess  
 */  
public abstract void slipRight(SlipEntity slipEntity,  
                                MotionEvent event, boolean isSuccess);  
  
/**  
 * Listen some dock(more than DOCK_L,DOCK_R), normally need not  
 * implement it unless more docks are needed.  
 *  
 * @param slipEntity  
 * @param event  
 * @param dockPer  
 */  
public abstract void slipDock(SlipEntity slipEntity, MotionEvent event,  
                             float dockPer);  
}  
  
public OnSlipListener getOnSlipListener() {  
    return onSlipListener;  
}
```

```
    public void setOnSlipListener(OnSlipListener onSlipListener) {  
        this.onSlipListener = onSlipListener;  
    }  
}
```

IaiaiActivity.java

```
package com.iaiai.activity;  
  
import android.app.Activity;  
import android.graphics.Bitmap;  
import android.graphics.drawable.BitmapDrawable;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.MotionEvent;  
  
import com.iaiai.activity.SlipView.OnSlipListener;  
  
/**  
 *  
 * <p>  
 * Title: IaiaiActivity.java  
 * </p>  
 * <p>  
 * E-Mail: 176291935@qq.com  
 * </p>  
 * <p>  
 * QQ: 176291935  
 * </p>  
 * <p>  
 * Http: iaiai.iteye.com  
 * </p>  
 * <p>  
 * Create time: 2011-8-11  
 * </p>
```

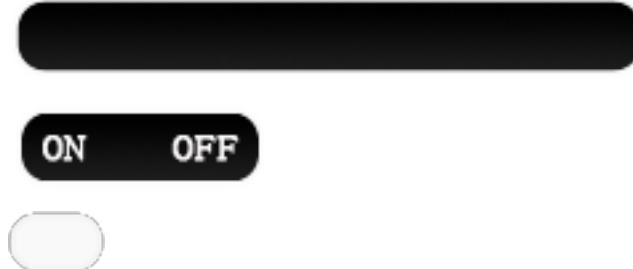
```
*  
* @author 丸子  
* @version 0.0.1  
*/  
  
public class IaiaiActivity extends Activity {  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        // setContentView(R.layout.main);  
  
        SlipView view;  
  
        // 1.Code  
        Bitmap backImage = ((BitmapDrawable) getResources().getDrawable(  
            R.drawable.back5)).getBitmap();  
        Bitmap frotImage = ((BitmapDrawable) getResources().getDrawable(  
            R.drawable.frot1)).getBitmap();  
        SlipEntity entity = new SlipEntity(backImage, frotImage);  
        entity.setCurDockPos(SlipEntity.DOCK_R);  
        entity.addDockPos((float) 0.5);  
  
        view = new SlipView(this, entity);  
  
        // 2.Code  
        // view = new SlipView(this);  
  
        // 3.XML  
        // view = (SlipView)this.findViewById(R.id.slipView);  
  
        setContentView(view);  
        view.setOnSlipListener(new OnSlipListener() {  
            @Override  
            public void slipLeft(SlipEntity slipEntity, MotionEvent event,  
                boolean isSuccess) {  
                Log.v("Left", Boolean.toString(isSuccess));  
            }  
        });  
    }  
}
```

```
    @Override
    public void slipRight(SlipEntity slipEntity, MotionEvent event,
                          boolean isSuccess) {
        Log.v("Right", Boolean.toString(isSuccess));
    }

    @Override
    public void slipDock(SlipEntity slipEntity, MotionEvent event,
                         float dockPer) {
        Log.v("Dock", "dockPer:" + dockPer);
    }
}

}
```

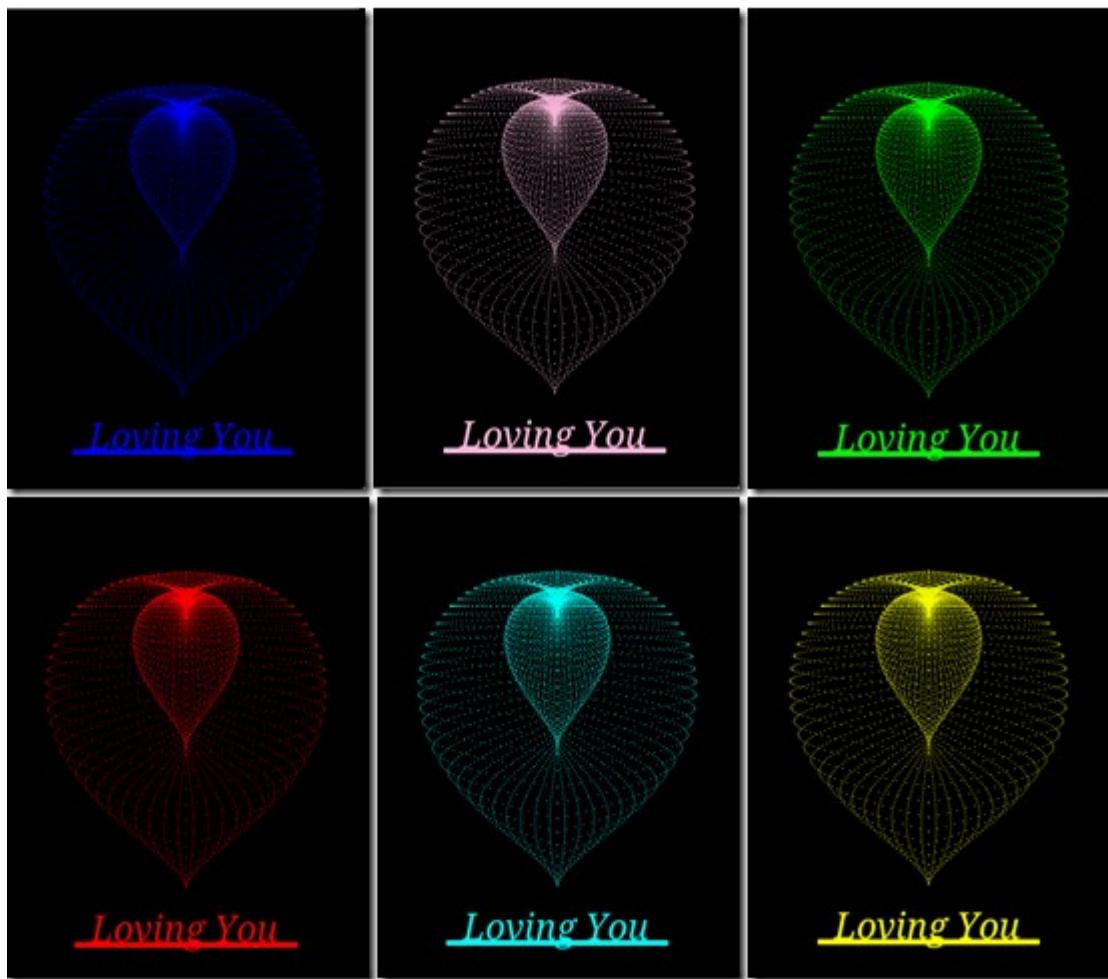
下面是用到的三张图片:



1.87 Android中利用画图类和线程画出闪烁的心形，送给亲爱的他

发表时间: 2011-08-12

大家先看图片：



因为前一段时间在写画图类，刚好有一个线程控制画图闪烁的，我就想说我能不能做一个心形闪烁的，出来的效果就如图，先贴再讲解代码：

里面设置两个类，一个是我们的activity类，这个类用来显示示图，然后建一个继承SurfaceView的类，我们在里面画图。先贴两个累的代码：

主类名：IaiaiMainActivity，画图类类名：Love.

```
package com.iaiai.activity;

import android.app.Activity;
import android.os.Bundle;

/**
 *
```

```
* <p>
* Title: IaiaiActivity.java
* </p>
* <p>
* E-Mail: 176291935@qq.com
* </p>
* <p>
* QQ: 176291935
* </p>
* <p>
* Http: iaiai.iteye.com
* </p>
* <p>
* Create time: 2011-8-11
* </p>
*
* @author 丸子
* @version 0.0.1
*/
public class IaiaiActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Love love = new Love(this);
        setContentView(love);
    }
}
```

```
package com.iaiai.activity;

import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
```

```
import android.graphics.Paint;
import android.graphics.RectF;
import android.graphics.Typeface;
import android.view.SurfaceHolder;
import android.view.SurfaceView;

/**
 *
 * <p>
 * Title: Love.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-8-11
 * </p>
 *
 * @author 丸子
 * @version 0.0.1
 */

public class Love extends SurfaceView implements SurfaceHolder.Callback,
    Runnable {

    boolean mbloop = false;
    SurfaceHolder mSurfaceHolder = null;
    private Canvas canvas;
    int miCount = 0;
    int y = 50;

    /**

```

```
* @param context
*/
public Love(Context context) {
    super(context);
    mSurfaceHolder = this.getHolder();
    mSurfaceHolder.addCallback(this);
    this.setFocusable(true);
    this.setKeepScreenOn(true);
    mbloop = true;
}

/*
 * (non-Javadoc)
 *
 * @see
 * android.view.SurfaceHolder.Callback#surfaceChanged(android.view.SurfaceHolder
 * , int, int, int)
 */
@Override
public void surfaceChanged(SurfaceHolder holder, int format, int width,
    int height) {
    // TODO Auto-generated method stub
}

/*
 * (non-Javadoc)
 *
 * @see
 * android.view.SurfaceHolder.Callback#surfaceCreated(android.view.SurfaceHolder
 * )
 */
@Override
public void surfaceCreated(SurfaceHolder holder) {
    // TODO Auto-generated method stub
    new Thread(this).start();
}
```

```
/*
 * (non-Javadoc)
 *
 *
 * @see android.view.SurfaceHolder.Callback#surfaceDestroyed(android.view.
 * SurfaceHolder)
 */
@Override
public void surfaceDestroyed(SurfaceHolder holder) {
    // TODO Auto-generated method stub
    mbloop = false;
}

/*
 * (non-Javadoc)
 *
 *
 * @see java.lang.Runnable#run()
 */
@Override
public void run() {
    // TODO Auto-generated method stub
    while (mbloop) {
        try {
            Thread.sleep(200);
        } catch (Exception e) {
            // TODO: handle exception
        }
        synchronized (mSurfaceHolder) {
            Draw();
        }
    }
}

private void Draw() {
    // TODO Auto-generated method stub
    canvas = mSurfaceHolder.lockCanvas();
```

```
try {
    if (mSurfaceHolder == null || canvas == null) {
        return;
    }
    if (miCount < 100) {
        miCount++;
    } else {
        miCount = 0;
    }
    Paint paint = new Paint();
    paint.setAntiAlias(true);
    paint.setColor(Color.BLACK);
    canvas.drawRect(0, 0, 320, 480, paint);
    switch (miCount % 6) {
        case 0:
            paint.setColor(Color.BLUE);
            break;
        case 1:
            paint.setColor(Color.GREEN);
            break;
        case 2:
            paint.setColor(Color.RED);
            break;
        case 3:
            paint.setColor(Color.YELLOW);
            break;
        case 4:
            paint.setColor(Color.argb(255, 255, 181, 216));
            break;
        case 5:
            paint.setColor(Color.argb(255, 0, 255, 255));
            break;
        default:
            paint.setColor(Color.WHITE);
            break;
    }
    int i, j;
```

```
        double x, y, r;

        for (i = 0; i <= 90; i++) {
            for (j = 0; j <= 90; j++) {
                r = Math.PI / 45 * i * (1 - Math.sin(Math.PI / 45 * j))
                    * 20;
                x = r * Math.cos(Math.PI / 45 * j)
                    * Math.sin(Math.PI / 45 * i) + 320 / 2;
                y = -r * Math.sin(Math.PI / 45 * j) + 400 / 4;
                canvas.drawPoint((float) x, (float) y, paint);
            }
        }

        paint.setTextSize(32);
        paint.setTypeface(Typeface.create(Typeface.SERIF, Typeface.ITALIC));

        RectF rect = new RectF(60, 400, 260, 405);
        canvas.drawRoundRect(rect, (float) 1.0, (float) 1.0, paint);
        canvas.drawText("Loving You", 75, 400, paint);
        mSurfaceHolder.unlockCanvasAndPost(canvas);
    } catch (Exception e) {
    }

}
}
```

关于这个程序要讲解的几点：

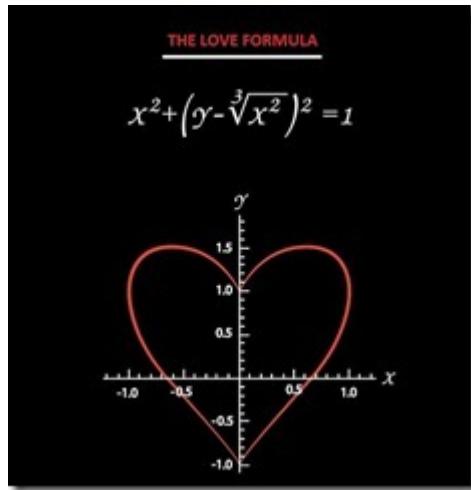
1. 画图的时候你可以继承View,也可以继承SurfaceView,这两者的区别在于：surfaceView是在一个新起的单独线程中可以重新绘制画面而View必须在UI的主线程中更新画面。SurfaceView可以控制表面的格式，比如大小，显示在屏幕中的位置，最关键是的提供了SurfaceHolder类，使用getHolder方法获取，还有涉及的surfaceCreated (SurfaceHolder holder),surfaceDestroyed(SurfaceHolder holder),surfaceChanged(SurfaceHolder holder, int format, int width, int height)方法，而在SurfaceHolder.Callback 接口回调中可以通过重写来改变这些方法

2.程序其实很简单，既然生命了Runnable接口，就有相对应的Run方法，在surfaceCreate（）的时候开启线程，线程每隔200ms就刷新一次，这样我们看到的效果就是闪烁的，每200毫秒 画一次图，根据经过的间隔时间来设置画笔的颜色，然后通过循环描点，画出心形，然后设置字体大小，画字和字下面的横线。

3.关于心形函数，是从一个例子中看来得，关于x和y的得到，

$x = r * \text{Math.cos}(\text{Math.PI} / 45 * j) * \text{Math.sin}(\text{Math.PI} / 45 * i) + 320 / 2; \quad y = -r * \text{Math.sin}(\text{Math.PI} / 45 * j) + 400 / 4;$

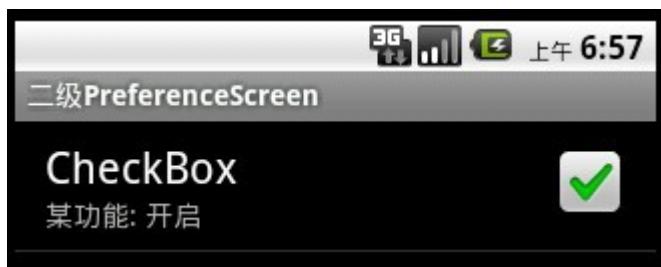
320是屏幕的宽度，本来竖屏我设置的是480，可是下面得写字，就设置为400的了，关于画更好看的心形还有一个函数，大家可以看下：



1.88 Android PreferenceActivity用法

发表时间: 2011-08-15

运行结果:



main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
    <PreferenceCategory android:title="PreferenceCategory 1">
        <CheckBoxPreference android:key="CheckBox1"
            android:title="CheckBox" android:summaryOn="某功能: 开启"
            android:summaryOff="某功能: 关闭" android:defaultValue="true" />
    </PreferenceCategory>
</PreferenceScreen>
```

```
<PreferenceCategory android:title="PreferenceCategory 2">
    <PreferenceScreen android:title="二级PreferenceScreen">
        <CheckBoxPreference android:key="CheckBox2"
            android:title="CheckBox" android:summaryOn="某功能: 开启"
            android:summaryOff="某功能: 关闭" android:defaultValue="true" />
    </PreferenceScreen>
</PreferenceCategory>
<PreferenceCategory android:title="PreferenceCategory 3">
    <ListPreference android:key="ListPreference"
        android:title="ListPreference" android:summary="ListPreference测试"
        android:dialogTitle="ListPreference" android:entries="@array/entries_li
        android:entryValues="@array/entriesvalue_list_preference" />
    <EditTextPreference android:key="EditTextPreference"
        android:title="EditTextPreference" android:summary="点击输入"
        android:dialogTitle="输入设置" />
    <RingtonePreference android:key="RingtonePreference"
        android:title="RingtonePreference" android:summary="选择铃声" />
</PreferenceCategory>
</PreferenceScreen>
```

array.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="entries_list_preference">
        <item>test1</item>
        <item>test2</item>
        <item>test3</item>
    </string-array>
    <string-array name="entriesvalue_list_preference">
        <item>1</item>
        <item>2</item>
        <item>3</item>
    </string-array>
</resources>
```

activity.java

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    addPreferencesFromResource(R.layout.main);
}
```

以上是PreferenceActivity的xml描述，那么在程序中我们只需要新建一个继承自PreferenceActivity的Activity，然后在主程序中调用就可以了。这个PreferenceActivity中的设置存储是完全自动的，你不需要再用代码去实现设置的存储，PreferenceActivity创建后会自动创建一个配置文件/data/data/you_package_name/shared_prefs/you_package_name_you_xml_name.xml。上例中自动生成的配置文件如下：

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<string name="EditTextPreference">12332312</string>
<string name="ListPreference">2</string>
<string name="RingtonePreference">content://settings/system/ringtone</string>
<boolean name="CheckBox1" value="true" />
<boolean name="CheckBox2" value="true" />
</map>
```

要取得其中的值可以通过如下的方法：

```
SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(this);
value = prefs.getString("ListPreference", "unset");
```

1.89 Android 9 PATCH 图片

发表时间: 2011-08-15

本文围绕 .9.png 格式图片讨论以下两个话题:

1. 该格式图片的特点
2. 制作方式

一 .9.png 格式的文件的特点

与传统的png 格式图片相比 , 9.png 格式图片在图片四周有一圈一个像素点组成的边沿 , 该边沿用于对图片的可扩展区和内容显示区进行定义。

这种格式的图片在android 环境下具有自适应调节大小的能力。

(1) 允许开发人员定义可扩展区域 , 当需要延伸图片以填充比图片本身更大区域时 , 可扩展区的内容被延伸。

(2) 允许开发人员定义内容显示区 , 用于显示文字或其他内容

如下图所示 :

左侧和上方的黑线交叉的部分即可扩展区域

右侧和下方的黑线交叉的部分即内容显示区

二、.9.png 图片的制作

android sdk 的 tools 文件夹下提供了制作该格式图片的工具 **draw9patch.bat**

使用此工具打开任意图片之后,将鼠标置于图片上,主界面如下图所示:



被黑色覆盖的是不可编辑(锁住)的区域,周围的一圈一个像素的边沿是可编辑区域。

按住鼠标左键 , 在左侧和上方的边沿画出可扩展区 , 在右侧和下方画出内容显示区 ,

完成绘制以后 , 选择file-> save ,即可保存为 .9.png 格式的文件,并在android 项目中使用。

1.90 Android 怎么样使用shape

发表时间: 2011-08-16

大家都应该明白的，在Android中常常会使用shape来定义控件的一些显示属性，那么怎么来用那，今天我们就来看一些shape的使用，大家看完这 篇以后就会对shape有了大体的了解，下面的是稍作总结请大家仔细的观看：

```
<shape>

<!-- 实心 -->
<solid android:color="#ff9d77"/>

<!-- 渐变 -->
<gradient
    android:startColor="#ff8c00"
    android:endColor="#FFFFFF"
    android:angle="270" />

<!-- 描边 -->
<stroke
    android:width="2dp"
    android:color="#dcfdcfc" />

<!-- 圆角 -->
<corners
    android:radius="2dp" />
<padding
    android:left="10dp"
    android:top="10dp"
    android:right="10dp"
    android:bottom="10dp" />

</shape>
```

solid : 实心，就是填充的意思

android:color指定填充的颜色

gradient : 渐变

android:startColor和android:endColor分别为起始和结束颜色，ndroid:angle是渐变角度，必须为45的整数倍。

另外渐变默认的模式为android:type="linear"，即线性渐变，可以指定渐变为径向渐变，
android:type="radial"，径向渐变需要指定半径android:gradientRadius="50"。

stroke : 描边

android:width="2dp" 描边的宽度，android:color 描边的颜色。

我们还可以把描边弄成虚线的形式，设置方式为：

android:dashWidth="5dp"

android:dashGap="3dp"

其中android:dashWidth表示'-'这样一个横线的宽度，android:dashGap表示之间隔开的距离。

corners : 圆角

android:radius为角的弧度，值越大角越圆。

我们还可以把四个角设定成不同的角度，方法为：

```
<corners
    android:topRightRadius="20dp" 右上角
    android:bottomLeftRadius="20dp" 右下角
    android:topLeftRadius="1dp" 左上角
    android:bottomRightRadius="0dp" 左下角
/>
```

这里有个地方需要注意，bottomLeftRadius是右下角，而不是左下角，这个有点郁闷，不过不影响使用，记得别搞错了就行。

还有网上看到有人说设置成0dp无效，不过我在测试中发现是可以的，我用的是2.2，可能修复了这个问题吧，如果无效的话那就只能设成1dp了。

padding : 间隔

这个就不用多说了，XML布局文件中经常用到。

大体的就是这样，以下是一个使用的基本示例：用在Selector中作为Button的背景，分别定义了按钮的一般状态、获得焦点状态和按下时的状态，具体代码如下：

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TestShapeButton"
    android:background="@drawable/button_selector"
/>>
```

我们来看看main.xml代码：

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

            <!-- 渐变 -->
            <gradient
                android:startColor="#ff8c00"
                android:endColor="#FFFFFF"
                android:type="radial"
                android:gradientRadius="50" />

            <!-- 描边 -->
            <stroke
                android:width="2dp"
                android:color="#dcdcdc"
                android:dashWidth="5dp"
                android:dashGap="3dp" />

            <!-- 圆角 -->
            <corners
                android:radius="2dp" />
        </shape>
    </item>
    <item android:state_focused="true" >
        <shape>
            <gradient
                android:startColor="#ff8c00"
                android:endColor="#FFFFFF"
                android:type="radial"
                android:gradientRadius="50" />
        </shape>
    </item>
    <item>
        <shape>
            <gradient
                android:startColor="#ff8c00"
                android:endColor="#FFFFFF"
                android:type="radial"
                android:gradientRadius="50" />
        </shape>
    </item>
</selector>
```

```
<padding
    android:left="10dp"
    android:top="10dp"
    android:right="10dp"
    android:bottom="10dp" />
</shape>

</item>

<item android:state_focused="true" >
<shape>
<gradient
    android:startColor="#ffc2b7"
    android:endColor="#ffc2b7"
    android:angle="270" />

<stroke
    android:width="2dp"
    android:color="#dcfdcfc" />
<corners
    android:radius="2dp" />

<padding
    android:left="10dp"
    android:top="10dp"
    android:right="10dp"
    android:bottom="10dp" />
</shape>
</item>

<item>
<shape>
<solid android:color="#ff9d77"/>
<stroke
    android:width="2dp"
    android:color="#fad3cf" />
```

```
<corners
    android:topRightRadius="5dp"
    android:bottomLeftRadius="5dp"
    android:topLeftRadius="0dp"
    android:bottomRightRadius="0dp"
/>
<padding
    android:left="10dp"
    android:top="10dp"
    android:right="10dp"
    android:bottom="10dp" />
</shape>
</item>
</selector>
```

1.91 Android 自定义PopupWindow实现的Menu(TabMenu)

发表时间: 2011-08-17

用过UCWEB-Android版的人都应该对其特殊的menu有印象，把menu做成Tab-Menu(支持分页的Menu)，可以容纳比Android传统的menu更丰富的内容(Android的menu超过6项则缩略在[更多]里)，本文参考网上的例子，对例子进行简化以及封装，使其作为一个复合控件融入自己的framework。

先来看看本文程序运行的效果：



TabMenu本身就是一个PopupWindow，PopupWindow上面放了两个GridView，第一个GridView就是分页标签，位于PopupWindow的顶部，第二个GridView是菜单，位于PopupWindow的主体。为了实现PopupWindow的弹出/退出的动画效果，本文使用了以下代码：

在工程的res文件夹里添加anim子目录，再新建文件popup_enter.xml：

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate android:fromYDelta="100%p" android:toYDelta="0" android:duration="1000" />
```

```
<alpha android:fromAlpha="0.0" android:toAlpha="1.0" android:duration="1000" />
</set>
```

新建文件popup_exit.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate android:fromYDelta="0" android:toYDelta="100%" android:duration="1000" />
    <alpha android:fromAlpha="1.0" android:toAlpha="0.0" android:duration="1000" />
</set>
```

在工程的values文件夹里新建文件popup_animation.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="PopupAnimation" parent="android:Animation">
        <item name="android:windowEnterAnimation">@anim/popup_enter</item>
        <item name="android:windowExitAnimation">@anim/popup_exit</item>
    </style>
</resources>
```

main.xml的源码如下 :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:id="@+id/LinearLayout01"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <TextView android:id="@+id/TextView01" android:layout_height="wrap_content"
        android:layout_width="fill_parent" android:text="扩展Menu"></TextView>
</LinearLayout>
```

TabMenu的封装类TabMenu.java的源码如下：

```
package com.iaiai.activity;

import android.content.Context;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.view.Gravity;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.GridView;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.PopupWindow;
import android.widget.TextView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.LinearLayout.LayoutParams;

/**
 *
 * <p>
 * Title: TabMenu.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-8-17
 * </p>
```

```
*  
* @author 丸子  
* @version 0.0.1  
*/  
  
public class TabMenu extends PopupWindow {  
    private GridView gvBody, gvTitle;  
    private LinearLayout mLayout;  
    private MenuTitleAdapter titleAdapter;  
  
    public TabMenu(Context context, OnItemClickListener titleClick,  
                  OnItemClickListener bodyClick, MenuTitleAdapter titleAdapter,  
                  int colorBgTabMenu, int aniTabMenu) {  
        super(context);  
  
        mLayout = new LinearLayout(context);  
        mLayout.setOrientation(LinearLayout.VERTICAL);  
        // 标题选项栏  
        gvTitle = new GridView(context);  
        gvTitle.setLayoutParams(new LayoutParams(LayoutParams.FILL_PARENT,  
                                              LayoutParams.WRAP_CONTENT));  
        gvTitle.setNumColumns(titleAdapter.getCount());  
        gvTitle.setStretchMode(GridView.STRETCH_COLUMN_WIDTH);  
        gvTitle.setVerticalSpacing(1);  
        gvTitle.setHorizontalSpacing(1);  
        gvTitle.setGravity(Gravity.CENTER);  
        gvTitle.setOnItemClickListener(titleClick);  
        gvTitle.setAdapter(titleAdapter);  
        gvTitle.setSelector(new ColorDrawable(Color.TRANSPARENT));// 选中的时候为透明色  
        this.titleAdapter = titleAdapter;  
        // 子选项栏  
        gvBody = new GridView(context);  
        gvBody.setLayoutParams(new LayoutParams(LayoutParams.FILL_PARENT,  
                                              LayoutParams.WRAP_CONTENT));  
        gvBody.setSelector(new ColorDrawable(Color.TRANSPARENT));// 选中的时候为透明色  
        gvBody.setNumColumns(4);  
        gvBody.setStretchMode(GridView.STRETCH_COLUMN_WIDTH);  
        gvBody.setVerticalSpacing(10);  
    }  
}
```

```
        gvBody.setHorizontalSpacing(10);
        gvBody.setPadding(10, 10, 10, 10);
        gvBody.setGravity(Gravity.CENTER);
        gvBody.setOnItemClickListener(bodyClick);
        mLayout.addView(gvTitle);
        mLayout.addView(gvBody);

        // 设置默认项
        this.setContentView(mLayout);
        this.setWidthLayoutParams.FILL_PARENT);
        this.setHeightLayoutParams.WRAP_CONTENT);
        this.setBackgroundDrawable(new ColorDrawable(colorBgTabMenu)); // 设置TabMenu菜单
        this.setAnimationStyle(aniTabMenu);
        this.setFocusable(true); // menu菜单获得焦点 如果没有获得焦点menu菜单中的控件事件无法
    }

    public void SetTitleSelect(int index) {
        gvTitle.setSelection(index);
        this.titleAdapter.SetFocus(index);
    }

    public void SetBodySelect(int index, int colorSelBody) {
        int count = gvBody.getChildCount();
        for (int i = 0; i < count; i++) {
            if (i != index)
                ((LinearLayout) gvBody.getChildAt(i))
                    .setBackgroundColor(Color.TRANSPARENT);
        }
        ((LinearLayout) gvBody.getChildAt(index))
            .setBackgroundColor(colorSelBody);
    }

    public void SetBodyAdapter(MenuBodyAdapter bodyAdapter) {
        gvBody.setAdapter(bodyAdapter);
    }

    /**

```

```
* 自定义Adapter , TabMenu的每个分页的主体
*
*/
static public class MenuBodyAdapter extends BaseAdapter {

    private Context mContext;
    private int fontColor, fontSize;
    private String[] texts;
    private int[] resID;

    /**
     * 设置TabMenu的分页主体
     *
     * @param context
     *          调用方的上下文
     * @param texts
     *          按钮集合的字符串数组
     * @param resID
     *          按钮集合的图标资源数组
     * @param fontSize
     *          按钮字体大小
     * @param color
     *          按钮字体颜色
     */
    public MenuBodyAdapter(Context context, String[] texts, int[] resID,
                          int fontSize, int fontColor) {
        this.mContext = context;
        this.fontColor = fontColor;
        this.texts = texts;
        this.fontSize = fontSize;
        this.resID = resID;
    }

    public int getCount() {
        return texts.length;
    }

    public Object getItem(int position) {
```

```
        return makeMenyBody(position);
    }

    public long getItemId(int position) {
        return position;
    }

    private LinearLayout makeMenyBody(int position) {
        LinearLayout result = new LinearLayout(this.mContext);
        result.setOrientation(LinearLayout.VERTICAL);
        result.setGravity(Gravity.CENTER_HORIZONTAL
                | Gravity.CENTER_VERTICAL);
        result.setPadding(10, 10, 10, 10);

        TextView text = new TextView(this.mContext);
        text.setText(texts[position]);
        text.setTextSize(fontSize);
        text.setTextColor(fontColor);
        text.setGravity(Gravity.CENTER);
        text.setPadding(5, 5, 5, 5);
        ImageView img = new ImageView(this.mContext);
        img.setImageResource(resID[position]);
        result.addView(img, new LinearLayout.LayoutParams(new LayoutParams(
                LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT)));
        result.addView(text);
        return result;
    }

    public View getView(int position, View convertView, ViewGroup parent) {
        return makeMenyBody(position);
    }

}

/**
 * 自定义Adapter,TabMenu的分页标签部分
 *
```

```
/*
static public class MenuTitleAdapter extends BaseAdapter {
    private Context mContext;
    private int fontColor, unselcolor, selcolor;
    private TextView[] title;

    /**
     * 设置TabMenu的title
     *
     * @param context
     *          调用方的上下文
     * @param titles
     *          分页标签的字符串数组
     * @param fontSize
     *          字体大小
     * @param fontcolor
     *          字体颜色
     * @param unselcolor
     *          未选中项的背景色
     * @param selcolor
     *          选中项的背景色
     */
    public MenuTitleAdapter(Context context, String[] titles, int fontSize,
                           int fontcolor, int unselcolor, int selcolor) {
        this.mContext = context;
        this.fontColor = fontcolor;
        this.unselcolor = unselcolor;
        this.selcolor = selcolor;
        this.title = new TextView[titles.length];
        for (int i = 0; i < titles.length; i++) {
            title[i] = new TextView(mContext);
            title[i].setText(titles[i]);
            title[i].setTextSize(fontSize);
            title[i].setTextColor(fontColor);
            title[i].setGravity(Gravity.CENTER);
            title[i].setPadding(10, 10, 10, 10);
        }
    }
}
```

```
}

public int getCount() {
    return title.length;
}

public Object getItem(int position) {
    return title[position];
}

public long getItemId(int position) {
    return title[position].getId();
}

/***
 * 设置选中的效果
 */
private void SetFocus(int index) {
    for (int i = 0; i < title.length; i++) {
        if (i != index) {
            title[i].setBackgroundDrawable(new ColorDrawable(unselcolor));
            title[i].setTextColor(fontColor);// 设置没选中项的字体颜色
        }
    }
    title[index].setBackgroundColor(0x00);// 设置选中项的颜色
    title[index].setTextColor(selcolor);// 设置选中项的字体颜色
}

public View getView(int position, View convertView, ViewGroup parent) {
    View v;
    if (convertView == null) {
        v = title[position];
    } else {
        v = convertView;
    }
    return v;
}
```

```
    }  
}
```

testTabMenu介绍了数据的定义以及TabMenu的使用，源码如下：

```
package com.iaiai.activity;  
  
import android.app.Activity;  
import android.graphics.Color;  
import android.os.Bundle;  
import android.view.Gravity;  
import android.view.Menu;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.AdapterView.OnItemClickListener;  
import android.widget.Toast;  
  
/**  
 *  
 * <p>  
 * Title: MainMenuView.java  
 * </p>  
 * <p>  
 * E-Mail: 176291935@qq.com  
 * </p>  
 * <p>  
 * QQ: 176291935  
 * </p>  
 * <p>  
 * Http: iaiai.iteye.com  
 * </p>  
 * <p>  
 * Create time: 2011-8-17  
 * </p>  
 */
```

```
* @author 丸子
* @version 0.0.1
*/
public class MainMenuView extends Activity {
    TabMenu.MenuBodyAdapter[] bodyAdapter = new TabMenu.MenuBodyAdapter[3];
    TabMenu.MenuTitleAdapter titleAdapter;
    TabMenu tabMenu;
    int selTitle = 0;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        // 设置分页栏的标题
        titleAdapter = new TabMenu.MenuTitleAdapter(this, new String[] { "常用",
                "设置", "工具" }, 16, 0xFF222222, Color.LTGRAY, Color.WHITE);
        // 定义每项分页栏的内容
        bodyAdapter[0] = new TabMenu.MenuBodyAdapter(this, new String[] {
                "常用1", "常用2", }, new int[] { R.drawable.menu_test,
                R.drawable.menu_bookmark }, 13, 0xFFFFFFFF);

        bodyAdapter[1] = new TabMenu.MenuBodyAdapter(this, new String[] {
                "设置1", "设置2", "设置3" }, new int[] { R.drawable.menu_edit,
                R.drawable.menu_delete, R.drawable.menu_fullscreen }, 13,
                0xFFFFFFFF);

        bodyAdapter[2] = new TabMenu.MenuBodyAdapter(this, new String[] {
                "工具1", "工具2", "工具3", "工具4" }, new int[] { R.drawable.menu_cut,
                R.drawable.menu_normalmode, R.drawable.menu_quit }, 13, 0xFFFFFFFF);

        tabMenu = new TabMenu(this, new TitleClickEvent(),
                new BodyClickEvent(), titleAdapter, 0x55123456,// TabMenu的背景
                R.style.PopupAnimation);// 出现与消失的动画

        tabMenu.update();
        tabMenu.SetTitleSelect(0);
```

```
        tabMenu.SetBodyAdapter(bodyAdapter[0]);  
    }  
  
    class TitleClickEvent implements OnItemClickListener {  
        @Override  
        public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,  
                long arg3) {  
            selTitle = arg2;  
            tabMenu.setTitleSelect(arg2);  
            tabMenu.SetBodyAdapter(bodyAdapter[arg2]);  
        }  
    }  
  
    class BodyClickEvent implements OnItemClickListener {  
        @Override  
        public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,  
                long arg3) {  
            tabMenu.SetBodySelect(arg2, Color.GRAY);  
            String str = "第" + String.valueOf(selTitle) + "栏\n\r" + "第"  
                    + String.valueOf(arg2) + "项";  
            Toast.makeText(MainMenuView.this, str, 500).show();  
        }  
    }  
  
    @Override  
    /**  
     * 创建MENU  
     */  
    public boolean onCreateOptionsMenu(Menu menu) {  
        menu.add("menu");// 必须创建一项  
        return super.onCreateOptionsMenu(menu);  
    }  
  
    @Override  
    /**
```

```
* 拦截MENU
*/
public boolean onMenuOpened(int featureId, Menu menu) {
    if (tabMenu != null) {
        if (tabMenu.isShowing())
            tabMenu.dismiss();
        else {
            tabMenu.showAtLocation(findViewById(R.id.LinearLayout01),
                Gravity.BOTTOM, 0, 0);
        }
    }
    return false;// 返回为true 则显示系统menu
}
}
```

1.92 Android 日历方式显示的日期选择组件

发表时间: 2011-08-19

运行结果:



此Widget组件只包含五个类文件:

- [DateWidget](#)
- [DateWidgetDayCell](#)
- [DateWidgetDayHeader](#)
- [DayStyle](#)
- [SymbolButton](#)

*红色为启动的activity

[DateWidget.java](#):

```
package com.iaiai;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;

import android.app.Activity;
```

```
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.LinearLayout.LayoutParams;
import android.widget.TextView;

/**
 *
 * <p>
 * Title: DateWidget.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-8-19
 * </p>
 *
 * @author 丸子
 * @version 0.0.1
 */
public class DateWidget extends Activity {
    private ArrayList<DateWidgetDayCell> days = new ArrayList<DateWidgetDayCell>();
    // private SimpleDateFormat dateMonth = new SimpleDateFormat("MMMM yyyy");
    private Calendar calStartDate = Calendar.getInstance();
    private Calendar calToday = Calendar.getInstance();
    private Calendar calCalendar = Calendar.getInstance();
    private Calendar calSelected = Calendar.getInstance();
    LinearLayout layContent = null;
```

```
Button btnPrev = null;
Button btnToday = null;
Button btnNext = null;
private int iFirstDayOfWeek = Calendar.MONDAY;
private int iMonthViewCurrentMonth = 0;
private int iMonthViewCurrentYear = 0;
public static final int SELECT_DATE_REQUEST = 111;
private static final int iDayCellSize = 38;
private static final int iDayHeaderHeight = 24;
private static final int iTotalWidth = (iDayCellSize * 7);
private TextView tv;
private int mYear;
private int mMonth;
private int mDay;

@Override
public void onCreate(Bundle icicle) {
    super.onCreate(icicle);
    iFirstDayOfWeek = Calendar.MONDAY;
    mYear = calSelected.get(Calendar.YEAR);
    mMonth = calSelected.get(Calendar.MONTH);
    mDay = calSelected.get(Calendar.DAY_OF_MONTH);
    setContentView(generateContentView());
    calStartDate = getCalendarStartDate();
    DateWidgetDayCell daySelected = updateCalendar();
    updateControlsState();
    if (daySelected != null)
        daySelected.requestFocus();
}

@Override
public void onStart() {
    super.onStart();
}

private LinearLayout createLayout(int iOrientation) {
```

```
LinearLayout lay = new LinearLayout(this);
lay.setLayoutParams(new LayoutParams(
        android.view.ViewGroup.LayoutParams.FILL_PARENT,
        android.view.ViewGroup.LayoutParams.WRAP_CONTENT));
lay.setOrientation(iOrientation);
return lay;
}

private Button createButton(String sText, int iWidth, int iHeight) {
    Button btn = new Button(this);
    btn.setText(sText);
    btn.setLayoutParams(new LayoutParams(iWidth, iHeight));
    return btn;
}

private void generateTopButtons(LinearLayout layTopControls) {
    final int iHorPadding = 24;
    final int iSmallButtonWidth = 60;
    btnToday = createButton("", iTotalWidth - iSmallButtonWidth
            - iSmallButtonWidth,
            android.view.ViewGroup.LayoutParams.WRAP_CONTENT);
    btnToday.setPadding(iHorPadding, btnToday.getPaddingTop(), iHorPadding,
            btnToday.getPaddingBottom());
    btnToday.setBackgroundResource(android.R.drawable.btn_default_small);

    SymbolButton btnPrev = new SymbolButton(this,
            SymbolButton.symbol.arrowLeft);
    btnPrev.setLayoutParams(new LayoutParams(iSmallButtonWidth,
            android.view.ViewGroup.LayoutParams.WRAP_CONTENT));
    btnPrev.setBackgroundResource(android.R.drawable.btn_default_small);

    SymbolButton btnNext = new SymbolButton(this,
            SymbolButton.symbol.arrowRight);
    btnNext.setLayoutParams(new LayoutParams(iSmallButtonWidth,
            android.view.ViewGroup.LayoutParams.WRAP_CONTENT));
    btnNext.setBackgroundResource(android.R.drawable.btn_default_small);
```

```
// set events

btnPrev.setOnClickListener(new Button.OnClickListener() {
    public void onClick(View arg0) {
        setPrevViewItem();
    }
});

btnToday.setOnClickListener(new Button.OnClickListener() {
    public void onClick(View arg0) {

        setTodayViewItem();
        String s = calToday.get(Calendar.YEAR) + "/"
            + (calToday.get(Calendar.MONTH) + 1);
        btnToday.setText(s);
    }
});

btnNext.setOnClickListener(new Button.OnClickListener() {
    public void onClick(View arg0) {
        setNextViewItem();
    }
});

layTopControls.setGravity(Gravity.CENTER_HORIZONTAL);
layTopControls.addView(btnPrev);
layTopControls.addView(btnToday);
layTopControls.addView(btnNext);

}

private View generateContentView() {
    LinearLayout layMain = createLayout(LinearLayout.VERTICAL);
    layMain.setPadding(8, 8, 8, 8);
    LinearLayout layTopControls = createLayout(LinearLayout.HORIZONTAL);

    layContent = createLayout(LinearLayout.VERTICAL);
    layContent.setPadding(20, 0, 20, 0);
    generateTopButtons(layTopControls);
    generateCalendar(layContent);
    layMain.addView(layTopControls);
}
```

```
    layMain.addView(layContent);

    tv = new TextView(this);
    layMain.addView(tv);
    return layMain;
}

private View generateCalendarRow() {
    LinearLayout layRow = createLayout(LinearLayout.HORIZONTAL);
    for (int iDay = 0; iDay < 7; iDay++) {
        DateWidgetDayCell dayCell = new DateWidgetDayCell(this,
                iDayCellSize, iDayCellSize);
        dayCell.setItemClick(mOnDayCellClick);
        days.add(dayCell);
        layRow.addView(dayCell);
    }
    return layRow;
}

private View generateCalendarHeader() {
    LinearLayout layRow = createLayout(LinearLayout.HORIZONTAL);
    for (int iDay = 0; iDay < 7; iDay++) {
        DateWidgetDayHeader day = new DateWidgetDayHeader(this,
                iDayCellSize, iDayHeaderHeight);
        final int iWeekDay = DayStyle.getWeekDay(iDay, iFirstDayOfWeek);
        day.setData(iWeekDay);
        layRow.addView(day);
    }
    return layRow;
}

private void generateCalendar(LinearLayout layContent) {
    layContent.addView(generateCalendarHeader());
    days.clear();
    for (int iRow = 0; iRow < 6; iRow++) {
        layContent.addView(generateCalendarRow());
    }
}
```

```
}

private Calendar getCalendarStartDate() {
    calToday.setTimeInMillis(System.currentTimeMillis());
    calToday.setFirstDayOfWeek(iFirstDayOfWeek);

    if (calSelected.getTimeInMillis() == 0) {
        calStartDate.setTimeInMillis(System.currentTimeMillis());
        calStartDate.setFirstDayOfWeek(iFirstDayOfWeek);
    } else {
        calStartDate.setTimeInMillis(calSelected.getTimeInMillis());
        calStartDate.setFirstDayOfWeek(iFirstDayOfWeek);
    }

    UpdateStartDateForMonth();

    return calStartDate;
}

private DateWidgetDayCell updateCalendar() {
    DateWidgetDayCell daySelected = null;
    boolean bSelected = false;
    final boolean bIsSelection = (calSelected.getTimeInMillis() != 0);
    final int iSelectedYear = calSelected.get(Calendar.YEAR);
    final int iSelectedMonth = calSelected.get(Calendar.MONTH);
    final int iSelectedDay = calSelected.get(Calendar.DAY_OF_MONTH);
    calCalendar.setTimeInMillis(calStartDate.getTimeInMillis());
    for (int i = 0; i < days.size(); i++) {
        final int iYear = calCalendar.get(Calendar.YEAR);
        final int iMonth = calCalendar.get(Calendar.MONTH);
        final int iDay = calCalendar.get(Calendar.DAY_OF_MONTH);
        final int iDayOfWeek = calCalendar.get(Calendar.DAY_OF_WEEK);
        DateWidgetDayCell dayCell = days.get(i);
        // check today
        boolean bToday = false;
        if (calToday.get(Calendar.YEAR) == iYear)
            if (calToday.get(Calendar.MONTH) == iMonth)
```

```
        if (calToday.get(Calendar.DAY_OF_MONTH) == iDay)
            bToday = true;

        // check holiday
        boolean bHoliday = false;
        if ((iDayOfWeek == Calendar.SATURDAY)
            || (iDayOfWeek == Calendar.SUNDAY))
            bHoliday = true;
        if ((iMonth == Calendar.JANUARY) && (iDay == 1))
            bHoliday = true;

        dayCell.setData(iYear, iMonth, iDay, bToday, bHoliday,
                        iMonthViewCurrentMonth);
        bSelected = false;
        if (bIsSelection)
            if ((iSelectedDay == iDay) && (iSelectedMonth == iMonth)
                && (iSelectedYear == iYear)) {
                bSelected = true;
            }
        dayCell.setSelected(bSelected);
        if (bSelected)
            daySelected = dayCell;
        calCalendar.add(Calendar.DAY_OF_MONTH, 1);
    }
    layContent.invalidate();
    return daySelected;
}

private void UpdateStartDateForMonth() {
    iMonthViewCurrentMonth = calStartDate.get(Calendar.MONTH);
    iMonthViewCurrentYear = calStartDate.get(Calendar.YEAR);
    calStartDate.set(Calendar.DAY_OF_MONTH, 1);
    UpdateCurrentMonthDisplay();
    // update days for week
    int iDay = 0;
    int iStartDay = iFirstDayOfWeek;
    if (iStartDay == Calendar.MONDAY) {
        iDay = calStartDate.get(Calendar.DAY_OF_WEEK) - Calendar.MONDAY;
```

```
        if (iDay < 0)
            iDay = 6;
    }
    if (iStartDay == Calendar.SUNDAY) {
        iDay = calStartDate.get(Calendar.DAY_OF_WEEK) - Calendar.SUNDAY;
        if (iDay < 0)
            iDay = 6;
    }
    calStartDate.add(Calendar.DAY_OF_WEEK, -iDay);
}

private void UpdateCurrentMonthDisplay() {
    String s = calCalendar.get(Calendar.YEAR) + "/"
        + (calCalendar.get(Calendar.MONTH) + 1); // dateMonth.format(cal)
    btnToday.setText(s);
    mYear = calCalendar.get(Calendar.YEAR);
}

private void setPrevViewItem() {
    iMonthViewCurrentMonth--;
    if (iMonthViewCurrentMonth == -1) {
        iMonthViewCurrentMonth = 11;
        iMonthViewCurrentYear--;
    }
    calStartDate.set(Calendar.DAY_OF_MONTH, 1);
    calStartDate.set(Calendar.MONTH, iMonthViewCurrentMonth);
    calStartDate.set(Calendar.YEAR, iMonthViewCurrentYear);
    UpdateStartDateForMonth();
    updateCalendar();
}

private void setTodayViewItem() {
    calToday.setTimeInMillis(System.currentTimeMillis());
    calToday.setFirstDayOfWeek(iFirstDayOfWeek);
    calStartDate.setTimeInMillis(calToday.getTimeInMillis());
    calStartDate.setFirstDayOfWeek(iFirstDayOfWeek);
```

```
        UpdateStartDateForMonth();
        updateCalendar();
    }

    private void setNextViewItem() {
        iMonthViewCurrentMonth++;
        if (iMonthViewCurrentMonth == 12) {
            iMonthViewCurrentMonth = 0;
            iMonthViewCurrentYear++;
        }
        calStartDate.set(Calendar.DAY_OF_MONTH, 1);
        calStartDate.set(Calendar.MONTH, iMonthViewCurrentMonth);
        calStartDate.set(Calendar.YEAR, iMonthViewCurrentYear);
        UpdateStartDateForMonth();
        updateCalendar();
    }

    private DateWidgetDayCell.OnItemClick mOnDayCellClick = new DateWidgetDayCell.OnItemClick() {
        public void OnClick(DateWidgetDayCell item) {
            calSelected.setTimeInMillis(item.getDate().getTimeInMillis());
            item.setSelected(true);
            updateCalendar();
            updateControlsState();
        }
    };
}

private void updateControlsState() {
    SimpleDateFormat dateFull = new SimpleDateFormat("d MMMM yyyy");
    mYear = calSelected.get(Calendar.YEAR);
    mMonth = calSelected.get(Calendar.MONTH);
    mDay = calSelected.get(Calendar.DAY_OF_MONTH);
    tv.setText(new StringBuilder().append(mYear).append("/")
              .append(format(mMonth + 1)).append("/").append(format(mDay))
              .append("----").append(dateFull.format(calSelected.getTime())));
}
```

```
private String format(int x) {  
    String s = "" + x;  
    if (s.length() == 1)  
        s = "0" + s;  
    return s;  
}  
}
```

DateWidgetDayCell.java:

```
package com.iaiai;  
  
import java.util.Calendar;  
  
import android.content.Context;  
import android.graphics.Canvas;  
import android.graphics.LinearGradient;  
import android.graphics.Paint;  
import android.graphics.Rect;  
import android.graphics.RectF;  
import android.graphics.Shader;  
import android.view.KeyEvent;  
import android.view.MotionEvent;  
import android.view.View;  
import android.view.animation.AlphaAnimation;  
import android.widget.LinearLayout.LayoutParams;  
  
/**  
 *  
 * <p>  
 * Title: DateWidgetDayCell.java  
 * </p>  
 * <p>  
 * E-Mail: 176291935@qq.com  
 */
```

```
* </p>
* <p>
* QQ: 176291935
* </p>
* <p>
* Http: iaiai.iteye.com
* </p>
* <p>
* Create time: 2011-8-19
* </p>
*
* @author 丸子
* @version 0.0.1
*/
public class DateWidgetDayCell extends View {
    // types
    public interface OnItemClick {
        public void OnClick(DateWidgetDayCell item);
    }

    public static int ANIM_ALPHA_DURATION = 100;
    // fields
    private final static float fTextSize = 22;
    private final static int iMargin = 1;
    private final static int iAlphaInactiveMonth = 0x88;

    // fields
    private int iDateYear = 0;
    private int iDateMonth = 0;
    private int iDateDay = 0;

    // fields
    private OnItemClick itemClick = null;
    private Paint pt = new Paint();
    private RectF rect = new RectF();
    private String sDate = "";
```

```
// fields

private boolean bSelected = false;
private boolean bIsActiveMonth = false;
private boolean bToday = false;
private boolean bHoliday = false;
private boolean bTouchedDown = false;

// methods

public DateWidgetDayCell(Context context, int iWidth, int iHeight) {
    super(context);
    setFocusable(true);
    setLayoutParams(new LayoutParams(iWidth, iHeight));
}

public boolean getSelected() {
    return this.bSelected;
}

@Override
public void setSelected(boolean bEnable) {
    if (this.bSelected != bEnable) {
        this.bSelected = bEnable;
        this.invalidate();
    }
}

public void setData(int iYear, int iMonth, int iDay, boolean bToday,
                   boolean bHoliday, int iActiveMonth) {
    iDateYear = iYear;
    iDateMonth = iMonth;
    iDateDay = iDay;

    this.sDate = Integer.toString(iDateDay);
    this.bIsActiveMonth = (iDateMonth == iActiveMonth);
    this.bToday = bToday;
    this.bHoliday = bHoliday;
}
```

```
public void setItemClick(OnItemClick itemClick) {  
    this.itemClick = itemClick;  
}  
  
private int getTextHeight() {  
    return (int) (-pt.ascent() + pt.descent());  
}  
  
@Override  
public boolean onKeyDown(int keyCode, KeyEvent event) {  
    boolean bResult = super.onKeyDown(keyCode, event);  
    if ((keyCode == KeyEvent.KEYCODE_DPAD_CENTER)  
        || (keyCode == KeyEvent.KEYCODE_ENTER)) {  
        doItemClick();  
    }  
    return bResult;  
}  
  
@Override  
public boolean onKeyUp(int keyCode, KeyEvent event) {  
    boolean bResult = super.onKeyUp(keyCode, event);  
    return bResult;  
}  
  
public void doItemClick() {  
    if (itemClick != null)  
        itemClick.OnClick(this);  
}  
  
@Override  
protected void onFocusChanged(boolean gainFocus, int direction,  
    Rect previouslyFocusedRect) {  
    super.onFocusChanged(gainFocus, direction, previouslyFocusedRect);  
    invalidate();  
}
```

```
public Calendar getDate() {
    Calendar calDate = Calendar.getInstance();
    calDate.clear();
    calDate.set(Calendar.YEAR, iDateYear);
    calDate.set(Calendar.MONTH, iDateMonth);
    calDate.set(Calendar.DAY_OF_MONTH, iDateDay);
    return calDate;
}

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

    // init rectangles
    rect.set(0, 0, this.getWidth(), this.getHeight());
    rect.inset(1, 1);

    // drawing
    final boolean bFocused = IsViewFocused();

    drawDayView(canvas, bFocused);
    drawDayNumber(canvas, bFocused);
}

private void drawDayView(Canvas canvas, boolean bFocused) {
    if (bSelected || bFocused) {
        LinearGradient lGradBkg = null;

        if (bFocused) {
            lGradBkg = new LinearGradient(rect.left, 0, rect.right, 0,
                DayStyle.iColorBkgFocusDark,
                DayStyle.iColorBkgFocusLight, Shader.TileMode.C
            }
        }

        if (bSelected) {
            lGradBkg = new LinearGradient(rect.left, 0, rect.right, 0,
                DayStyle.iColorBkgSelectedDark,
```

```
DayStyle.iColorBkgSelectedLight, Shader.TileMode.BLOCK;
```

```
        }
```

```
        if (lGradBkg != null) {
            pt.setShader(lGradBkg);
            canvas.drawRect(rect, pt);
        }

        pt.setShader(null);

    } else {

        pt.setColor(DayStyle.getColorBkg(bHoliday, bToday));
        if (!bIsActiveMonth)
            pt.setAlpha(iAlphaInactiveMonth);
        canvas.drawRect(rect, pt);
    }
}
```

```
public void drawDayNumber(Canvas canvas, boolean bFocused) {
    // draw day number
    pt.setTypeface(null);
    pt.setAntiAlias(true);
    pt.setShader(null);
    pt.setFakeBoldText(true);
    pt.setTextSize(fTextSize);

    pt.setUnderlineText(false);
    if (bToday)
        pt.setUnderlineText(true);

    int iTTextPosX = (int) rect.right - (int) pt.measureText(sDate);
    int iTTextPosY = (int) rect.bottom + (int) (-pt.ascent())
        - getTextHeight();

    iTTextPosX -= ((int) rect.width() >> 1)
        - ((int) pt.measureText(sDate) >> 1);
}
```

```
iTextPosY -= ((int) rect.height() >> 1) - (getTextHeight() >> 1);

// draw text
if (bSelected || bFocused) {
    if (bSelected)
        pt.setColor(DayStyle.iColorTextSelected);
    if (bFocused)
        pt.setColor(DayStyle.iColorTextFocused);
} else {
    pt.setColor(DayStyle.getColorText(bHoliday, bToday));
}

if (!bIsActiveMonth)
    pt.setAlpha(iAlphaInactiveMonth);

canvas.drawText(sDate, iTextPosX, iTextPosY + iMargin, pt);

pt.setUnderlineText(false);
}

public boolean IsViewFocused() {
    return (this.isFocused() || bTouchedDown);
}

@Override
public boolean onTouchEvent(MotionEvent event) {
    boolean bHandled = false;
    if (event.getAction() == MotionEvent.ACTION_DOWN) {
        bHandled = true;
        bTouchedDown = true;
        invalidate();
        startAlphaAnimIn(DateWidgetDayCell.this);
    }
    if (event.getAction() == MotionEvent.ACTION_CANCEL) {
        bHandled = true;
        bTouchedDown = false;
        invalidate();
    }
}
```

```
        }

        if (event.getAction() == MotionEvent.ACTION_UP) {
            bHandled = true;
            bTouchedDown = false;
            invalidate();
            doItemClick();
        }

        return bHandled;
    }

    public static void startAlphaAnimIn(View view) {
        AlphaAnimation anim = new AlphaAnimation(0.5F, 1);
        anim.setDuration(ANIM_ALPHA_DURATION);
        anim.startNow();
        view.startAnimation(anim);
    }

}
```

DateWidgetDayHeader.java:

```
package com.iaiai;

import java.util.Calendar;

import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Paint;
import android.graphics.RectF;
import android.view.View;

import android.widget.LinearLayout.LayoutParams;

/**
 *
 * <p>
```

```
* Title: DateWidgetDayHeader.java
*
* </p>
*
* <p>
*
* E-Mail: 176291935@qq.com
*
* </p>
*
* <p>
*
* QQ: 176291935
*
* </p>
*
* <p>
*
* Http: iaiai.iteye.com
*
* </p>
*
* <p>
*
* Create time: 2011-8-19
*
* </p>
*
*
* @author 丸子
*
* @version 0.0.1
*/
public class DateWidgetDayHeader extends View {
    // fields
    private final static int iDayHeaderFontSize = 12;

    // fields
    private Paint pt = new Paint();
    private RectF rect = new RectF();
    private int iWeekDay = -1;
    private boolean bHoliday = false;

    // methods
    public DateWidgetDayHeader(Context context, int iWidth, int iHeight) {
        super(context);
        setLayoutParams(new LayoutParams(iWidth, iHeight));
    }

    public void setData(int iWeekDay) {
        this.iWeekDay = iWeekDay;
        this.bHoliday = false;
    }
}
```

```
if ((iWeekDay == Calendar.SATURDAY) || (iWeekDay == Calendar.SUNDAY))
    this.bHoliday = true;
}

private void drawDayHeader(Canvas canvas) {
    if (iWeekDay != -1) {
        // background
        pt.setColor(DayStyle.getColorFrameHeader(bHoliday));
        canvas.drawRect(rect, pt);

        // text
        pt.setTypeface(null);
        pt.setTextSize(iDayHeaderFontSize);
        pt.setAntiAlias(true);
        pt.setFakeBoldText(true);
        pt.setColor(DayStyle.getColorTextHeader(bHoliday));

        final int iTextPosY = getTextHeight();
        final String sDayName = DayStyle.getWeekDayName(iWeekDay);

        // draw day name
        final int iDayNamePosX = (int) rect.left
            + ((int) rect.width() >> 1)
            - ((int) pt.measureText(sDayName) >> 1);
        canvas.drawText(sDayName, iDayNamePosX, rect.top + iTextPosY + 2,
            pt);
    }
}

private int getTextHeight() {
    return (int) (-pt.ascent() + pt.descent());
}

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

```
        // init rectangles
        rect.set(0, 0, this.getWidth(), this.getHeight());
        rect.inset(1, 1);

        // drawing
        drawDayHeader(canvas);
    }

}
```

DayStyle.java:

```
package com.iaiai;

import java.util.Calendar;

/**
 *
 * <p>
 * Title: DayStyle.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-8-19
 * </p>
 *
 * @author 丸子

```

```
* @version 0.0.1
*/
public class DayStyle {
    // methods

    private static String[] getWeekDayNames() {
        String[] vec = new String[10];
        // vec[Calendar.SUNDAY] = "Sun";
        // vec[Calendar.MONDAY] = "Mon";
        // vec[Calendar.TUESDAY] = "Tue";
        // vec[Calendar.WEDNESDAY] = "Wed";
        // vec[Calendar.THURSDAY] = "Thu";
        // vec[Calendar.FRIDAY] = "Fri";
        // vec[Calendar.SATURDAY] = "Sat";

        vec[Calendar.SUNDAY] = "日";
        vec[Calendar.MONDAY] = "月";
        vec[Calendar.TUESDAY] = "火";
        vec[Calendar.WEDNESDAY] = "水";
        vec[Calendar.THURSDAY] = "木";
        vec[Calendar.FRIDAY] = "金";
        vec[Calendar.SATURDAY] = "土";
        return vec;
    }

    public static String getWeekDayName(int iDay) {
        return vecStrWeekDayNames[iDay];
    }

    // fields
    private final static String[] vecStrWeekDayNames = getWeekDayNames();

    // fields
    public final static int iColorFrameHeader = 0xff666666;
    public final static int iColorFrameHeaderHoliday = 0xff707070;
    public final static int iColorTextHeader = 0xffcccccc;
    public final static int iColorTextHeaderHoliday = 0xffd0d0d0;
```

```
public final static int iColorText = 0xffdddddd;
public final static int iColorBkg = 0xff888888;
public final static int iColorTextHoliday = 0xffff0f0f0;
public final static int iColorBkgHoliday = 0xffaaaaaa;

public final static int iColorTextToday = 0xff002200;
public final static int iColorBkgToday = 0xff88bb88;

public final static int iColorTextSelected = 0xff001122;
public final static int iColorBkgSelectedLight = 0xffbbddff;
public final static int iColorBkgSelectedDark = 0xff225599;

public final static int iColorTextFocused = 0xff221100;
public final static int iColorBkgFocusLight = 0xfffffddbb;
public final static int iColorBkgFocusDark = 0xffaa5500;

// methods

public static int getColorFrameHeader(boolean bHoliday) {
    if (bHoliday)
        return iColorFrameHeaderHoliday;
    return iColorFrameHeader;
}

public static int getColorTextHeader(boolean bHoliday) {
    if (bHoliday)
        return iColorTextHeaderHoliday;
    return iColorTextHeader;
}

public static int getColorText(boolean bHoliday, boolean bToday) {
    if (bToday)
        return iColorTextToday;
    if (bHoliday)
        return iColorTextHoliday;
    return iColorText;
}
```

```
public static int getColorBkg(boolean bHoliday, boolean bToday) {  
    if (bToday)  
        return iColorBkgToday;  
    if (bHoliday)  
        return iColorBkgHoliday;  
    return iColorBkg;  
}  
  
public static int getWeekDay(int index, int iFirstDayOfWeek) {  
    int iWeekDay = -1;  
  
    if (iFirstDayOfWeek == Calendar.MONDAY) {  
        iWeekDay = index + Calendar.MONDAY;  
        if (iWeekDay > Calendar.SATURDAY)  
            iWeekDay = Calendar.SUNDAY;  
    }  
  
    if (iFirstDayOfWeek == Calendar.SUNDAY) {  
        iWeekDay = index + Calendar.SUNDAY;  
    }  
  
    return iWeekDay;  
}  
}
```

SymbolButton.java:

```
package com.iaiai;  
  
import android.content.Context;  
import android.graphics.Canvas;  
import android.graphics.Paint;  
import android.graphics.RectF;
```

```
import android.widget.Button;

/**
 *
 * <p>
 * Title: SymbolButton.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-8-19
 * </p>
 *
 * @author 丸子
 * @version 0.0.1
 */

public class SymbolButton extends Button {
    // fields

    private static final int iColor = 0xffaaaaaa;
    private static final int iColorActive = 0xff442200;

    // fields
    public enum symbol {
        none, arrowLeft, arrowRight
    };

    // fields
    private Paint pt = new Paint();
    private RectF rect = new RectF();
    private RectF rectDraw = new RectF();
```

```
private symbol symbolType = symbol.none;

// methods
public SymbolButton(Context context, symbol symbolType) {
    super(context);
    this.symbolType = symbolType;
}

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

    pt.setAntiAlias(true);
    pt.setStrokeCap(Paint.Cap.ROUND);

    rectDraw.set(0, 0, getWidth(), getHeight());
    rectDraw.left += 6;
    rectDraw.right -= 6;
    rectDraw.top += 4;
    rectDraw.bottom -= 8;

    if (symbolType != symbol.none) {
        pt.setStrokeWidth(5);

        pt.setColor(iColor);
        if (this.isPressed() || this.isFocused())
            pt.setColor(iColorActive);

        drawArrow(canvas);
    }
}

private void drawArrow(Canvas canvas) {
    rect.set(rectDraw);
    rect.inset(15, 5);
    canvas.drawLine(rect.left, rect.centerY(), rect.right, rect.centerY(),
        pt);
```

```
    if (symbolType == symbol.arrowRight) {
        canvas.drawLine(rect.right, rect.centerY(), rect.right - 6,
                        rect.top, pt);
        canvas.drawLine(rect.right, rect.centerY(), rect.right - 6,
                        rect.bottom, pt);
    }
    if (symbolType == symbol.arrowLeft) {
        canvas.drawLine(rect.left, rect.centerY(), rect.left + 6, rect.top,
                        pt);
        canvas.drawLine(rect.left, rect.centerY(), rect.left + 6,
                        rect.bottom, pt);
    }
}
```

1.93 Android 多层树完美实现

发表时间: 2011-08-20

效果:



main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="fill_parent"
    android:layout_height="wrap_content">
    <ImageView android:id="@+id/icon" android:layout_width="wrap_content"
        android:layout_height="fill_parent" android:layout_marginRight="6.0dip"
        android:layout_alignParentTop="true" android:layout_alignParentBottom="true" />
    <!--<TextView android:id="@+id/text" android:layout_gravity="center_vertical"
        android:layout_weight="1.0" android:layout_height="wrap_content" android:layout_width="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium" android:singleLine="true"
        android:textColor="#ffffffff" /> -->
    <TextView android:textAppearance="?android:attr/textAppearanceMedium" />

```

```
        android:gravity="center_vertical" android:id="@+id/text"
        android:textColor="#ffffff" android:layout_width="fill_parent"
        android:layout_height="fill_parent" android:singleLine="true"
        android:layout_toRightOf="@+id/icon" android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true" />
</RelativeLayout>
```

TreeElement.java:

```
package com.iaiai;

import java.util.ArrayList;

/**
 *
 * <p>
 * Title: TreeElement.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-8-20
 * </p>
 *
 * @author 丸子
 * @version 0.0.1
 */
public class TreeElement {
```

```
private String id;
private String outlineTitle;
private boolean mhasParent;
private boolean mhasChild;
private TreeElement parent;
private int level;
private ArrayList<TreeElement> childList = new ArrayList<TreeElement>();

public String getId() {
    return id;
}

public void setId(String id) {
    this.id = id;
}

public String getOutlineTitle() {
    return outlineTitle;
}

public void setOutlineTitle(String outlineTitle) {
    this.outlineTitle = outlineTitle;
}

public boolean isMhasParent() {
    return mhasParent;
}

public void setMhasParent(boolean mhasParent) {
    this.mhasParent = mhasParent;
}

public boolean isMhasChild() {
    return mhasChild;
}
```

```
public void setMhasChild(boolean mhasChild) {
    this.mhasChild = mhasChild;
}

public int getLevel() {
    return level;
}

public void setLevel(int level) {
    this.level = level;
}

public boolean isExpanded() {
    return expanded;
}

public void setExpanded(boolean expanded) {
    this.expanded = expanded;
}

public ArrayList<TreeElement> getChildList() {
    return childList;
}

public TreeElement getParent() {
    return parent;
}

public void setParent(TreeElement parent) {
    this.parent = parent;
}

// private OutlineElement outlineElement;
private boolean expanded;

public void addChild(TreeElement c) {
    this.childList.add(c);
```

```
        this.mhasParent = false;
        this.mhasChild = true;
        c.parent = this;
        c.level = this.level + 1;

    }

public TreeElement(String id, String title) {
    super();
    this.id = id;
    this.outlineTitle = title;
    this.level = 0;
    this.mhasParent = true;
    this.mhasChild = false;
    this.parent = null;
}

public TreeElement(String id, String outlineTitle, boolean mhasParent,
                   boolean mhasChild, TreeElement parent, int level, boolean expanded) {
    super();
    this.id = id;
    this.outlineTitle = outlineTitle;
    this.mhasParent = mhasParent;
    this.mhasChild = mhasChild;
    this.parent = parent;
    if (parent != null) {
        this.parent.getChildList().add(this);
    }
    this.level = level;
    this.expanded = expanded;
}

}
```

TreeView.java:

```
package com.iaiai;

import java.util.ArrayList;
import java.util.List;

import android.app.ListActivity;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

/**
 * <p>
 * Title: TreeView.java
 * </p>
 * <p>
 * E-Mail: 176291935@qq.com
 * </p>
 * <p>
 * QQ: 176291935
 * </p>
 * <p>
 * Http: iaiai.iteye.com
 * </p>
 * <p>
 * Create time: 2011-8-20
 * </p>
```

```
*  
* @author 丸子  
* @version 0.0.1  
*/  
  
public class TreeView extends ListActivity {  
    private ArrayList<TreeElement> mPdfOutlinesCount = new ArrayList<TreeElement>();  
    private TreeViewAdapter treeViewAdapter = null;  
  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        initData();  
        treeViewAdapter = new TreeViewAdapter(this, R.layout.main,  
                mPdfOutlinesCount);  
        setListAdapter(treeViewAdapter);  
        registerForContextMenu(getListView());  
    }  
  
    private void initData() {  
        TreeElement element1 = new TreeElement("01", "关键类");  
        TreeElement element2 = new TreeElement("02", "应用程序组件");  
        TreeElement element3 = new TreeElement("03", "Activity和任务");  
        TreeElement element4 = new TreeElement("04", "激活组件：intent");  
        TreeElement element5 = new TreeElement("05", "关闭组件");  
        TreeElement element6 = new TreeElement("06", "manifest文件");  
        TreeElement element7 = new TreeElement("07", "Intent过滤器");  
        TreeElement element8 = new TreeElement("08", "Affinity(吸引力)和新任务");  
        TreeElement element9 = new TreeElement("09", "加载模式");  
        TreeElement element10 = new TreeElement("10", "加载模式孩子1");  
        TreeElement element11 = new TreeElement("11", "加载模式孩子2");  
        TreeElement element12 = new TreeElement("12", "加载模式孩子2的孩子1");  
        TreeElement element13 = new TreeElement("13", "加载模式孩子2的孩子2");  
        TreeElement element14 = new TreeElement("14", "加载模式孩子1的孩子1");  
        TreeElement element15 = new TreeElement("15", "加载模式孩子1的孩子2");  
        TreeElement element16 = new TreeElement("16", "加载模式孩子1的孩子3");  
    }  
}
```

```
TreeElement element17 = new TreeElement("17", "加载模式孩子1的孩子4");
TreeElement element18 = new TreeElement("18", "加载模式孩子1的孩子5");
TreeElement element19 = new TreeElement("19", "加载模式孩子1的孩子6");

mPdfOutlinesCount.add(element1);
mPdfOutlinesCount.add(element2);
mPdfOutlinesCount.add(element9);
element2.addChild(element3);
element1.addChild(element4);
element1.addChild(element5);
element1.addChild(element6);
element1.addChild(element7);
element7.addChild(element8);
element9.addChild(element10);
element9.addChild(element11);
element11.addChild(element12);
element11.addChild(element13);
element10.addChild(element14);
element10.addChild(element15);
element10.addChild(element16);
element10.addChild(element17);
element10.addChild(element18);
element10.addChild(element19);

}

@SuppressWarnings("unchecked")
private class TreeViewAdapter extends ArrayAdapter {
    public TreeViewAdapter(Context context, int textViewResourceId,
                           List objects) {
        super(context, textViewResourceId, objects);
        mInflater = LayoutInflater.from(context);
        mfilelist = objects;
        mIconCollapse = BitmapFactory.decodeResource(
                context.getResources(), R.drawable.outline_list_collaps
        mIconExpand = BitmapFactory.decodeResource(context.getResources(),
                R.drawable.outline_list_expand);
    }
}
```

```
}

private LayoutInflator mInflater;
private List<TreeElement> mfilelist;
private Bitmap mIconCollapse;
private Bitmap mIconExpand;

public int getCount() {
    return mfilelist.size();
}

public Object getItem(int position) {
    return position;
}

public long getItemId(int position) {
    return position;
}

public View getView(int position, View convertView, ViewGroup parent) {
    ViewHolder holder;
    /* if (convertView == null) { */
    convertView = mInflater.inflate(R.layout.main, null);
    holder = new ViewHolder();
    holder.text = (TextView) convertView.findViewById(R.id.text);
    holder.icon = (ImageView) convertView.findViewById(R.id.icon);
    convertView.setTag(holder);
    /*
     * } else { holder = (ViewHolder) convertView.getTag(); }
     */
    final TreeElement obj = mfilelist.get(position);

    holder.text.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
```

```
        Log.i("TreeView", "obj.id:" + obj.getId());
    }
});

int level = obj.getLevel();
holder.icon.setPadding(25 * (level + 1),
                      holder.icon.getPaddingTop(), 0,
                      holder.icon.getPaddingBottom());
holder.text.setText(obj.getOutlineTitle());
if (obj.isMhasChild() && (obj.isExpanded() == false)) {
    holder.icon.setImageResource(mIconCollapse);
} else if (obj.isMhasChild() && (obj.isExpanded() == true)) {
    holder.icon.setImageResource(mIconExpand);
} else if (!obj.isMhasChild()) {
    holder.icon.setImageResource(mIconCollapse);
    holder.icon.setVisibility(View.INVISIBLE);
}
return convertView;
}

class ViewHolder {
    TextView text;
    ImageView icon;
}

@Override
protected void onListItemClick(ListView l, View v, int position, long id) {
    super.onListItemClick(l, v, position, id);
    Log.i("TreeView", "position:" + position);
    if (!mPdfOutlinesCount.get(position).isMhasChild()) {
        Toast.makeText(this, mPdfOutlinesCount.get(position)
                      .getOutlineTitle(), 2000);
        /*
         * int pageNumber; Intent i = getIntent(); element element =
         * mPdfOutlinesCount .get(position); pageNumber =
         */
    }
}
```

```
        * element.getOutlineElement().pageNumber; if (pageNumber <= 0) {  
        * String name = element.getOutlineElement().destName; pageNumber =  
        * idocviewer.getPageNumberForNameForOutline(name);  
        * element.getOutlineElement().pageNumber = pageNumber;  
        * element.getOutlineElement().destName = null; }  
        * i.putExtra("PageNumber", pageNumber); setResult(RESULT_OK, i);  
        * finish();  
        */  
  
    return;  
}  
  
if (mPdfOutlinesCount.get(position).isExpanded()) {  
    mPdfOutlinesCount.get(position).setExpanded(false);  
    TreeElement element = mPdfOutlinesCount.get(position);  
    ArrayList<TreeElement> temp = new ArrayList<TreeElement>();  
  
    for (int i = position + 1; i < mPdfOutlinesCount.size(); i++) {  
        if (element.getLevel() >= mPdfOutlinesCount.get(i).getLevel())  
            break;  
    }  
    temp.add(mPdfOutlinesCount.get(i));  
}  
  
mPdfOutlinesCount.removeAll(temp);  
  
treeViewAdapter.notifyDataSetChanged();  
/*  
 * fileExploreAdapter = new TreeViewAdapter(this, R.layout.outline,  
 * mPdfOutlinesCount);  
 */  
  
// setListAdapter(fileExploreAdapter);  
  
} else {  
    TreeElement obj = mPdfOutlinesCount.get(position);  
    obj.setExpanded(true);  
}
```

```
        int level = obj.getLevel();
        int nextLevel = level + 1;

        for (TreeElement element : obj.getChildList()) {
            element.setLevel(nextLevel);
            element.setExpanded(false);
            mPdfOutlinesCount.add(position + 1, element);

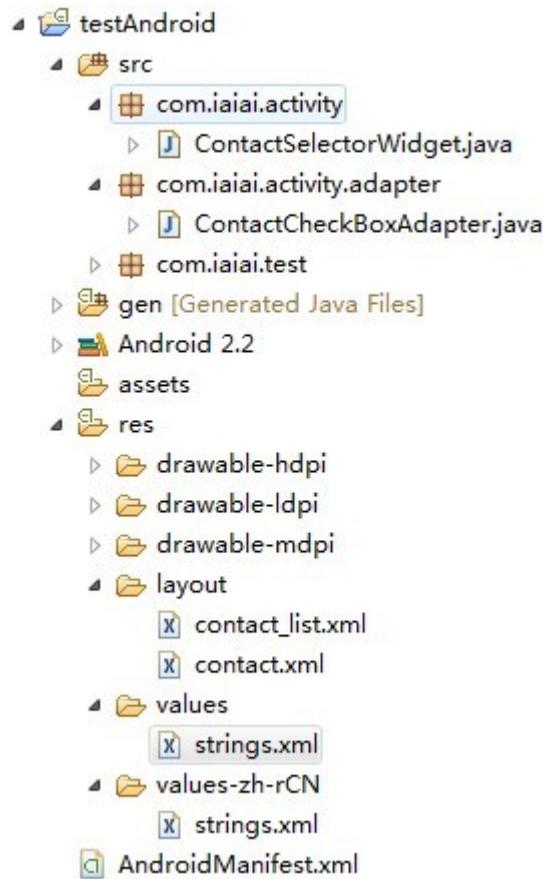
        }
        treeViewAdapter.notifyDataSetChanged();
        /*
         * fileExploreAdapter = new TreeViewAdapter(this, R.layout.outline,
         * mPdfOutlinesCount);
         */
        // setListAdapter(fileExploreAdapter);
    }
}

}
```

2.1 Android 联系人选择Widget

发表时间: 2011-06-02 关键字: Android, Java, ITeye, XML, 数据结构

先看一下目录结构:



AndroidManifest.xml配置文件:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.iaiai.activity" android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name="ContactSelectorWidget"
            android:label="@string/contact_title">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

```
</application>

<!-- 读取联系人 -->
<uses-permission android:name="android.permission.READ_CONTACTS" />

<!-- 写入联系人 -->
<uses-permission android:name="android.permission.WRITE_CONTACTS" />

<!-- 在SD卡中创建文件与删除文件权限 -->
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />

<!-- 往SD卡中写入数据权限 -->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

<!-- 电话拨号 -->
<uses-permission android:name="android.permission.CALL_PHONE" />

<!-- 访问网络 -->
<uses-permission android:name="android.permission.INTERNET" />

<!-- 访问Wi-Fi网络 -->
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />

</manifest>
```

values/strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">iaiai - Contact</string>
    <string name="contact_title">iaiai - Contact</string>
</resources>
```

values-zh-rCN/strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">丸子联系人</string>
    <string name="contact_title">丸子联系人</string>
</resources>
```

contact.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <ListView android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:id="@+id/contactListView">
    </ListView>
</LinearLayout>
```

contact_list.xml布局文件:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:id="@+id/contactItems"
    android:paddingBottom="5dip" android:paddingLeft="12dip"
    android:paddingTop="5dip" android:paddingRight="12dip">
    <ImageView android:id="@+id/contact_img"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"/>
    <TextView android:layout_height="wrap_content"
        android:layout_width="wrap_content" android:id="@+id/contact_name"
        android:textSize="20dip" android:paddingTop="12px"
        android:layout_marginLeft="70px">
    </TextView>
    <CheckBox android:id="@+id/contact_checkbox"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_alignParentRight="true" android:focusable="false"/>
</RelativeLayout>
```

ContactSelectorWidget.java类:

```
package com.iaiai.activity;

import java.io.InputStream;
```

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import android.app.Activity;
import android.content.ContentResolver;
import android.content.ContentUris;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.ListView;
import android.widget.Toast;

import com.iaiai.activity.adapter.ContactCheckBoxAdapter;

public class ContactSelectorWidget extends Activity {

    public static final String CONTACT_NAME = "NAME"; // 保存的名字

    public static final String CONTACT_ID = "ID"; // ID

    public static final String CONTACT_FACE = "FACE"; // 头像

    private ContactCheckBoxAdapter contactCheckBoxAdapter; // 配置适配器

    private ListView list;

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

```
        setContentView(R.layout.contact);

        list = (ListView) findViewById(R.id.contactListView);

        // 配置适配器
        contactCheckBoxAdapter = new ContactCheckBoxAdapter(this, getData()); // 布局里
        // 添加并且显示
        list.setAdapter(contactCheckBoxAdapter);

    }

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    menu.add(Menu.NONE, 1, 0, android.R.string.ok);
    menu.add(Menu.NONE, 2, 0, android.R.string.cancel);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case 1:
            Toast.makeText(
                this,
                "selector:" +
                contactCheckBoxAdapter.onSelectorData
            ).show();
            break;
        case 2:
            Toast.makeText(this, android.R.string.cancel, Toast.LENGTH_LONG)
                .show();
            break;
        default:
            break;
    }
    return super.onOptionsItemSelected(item);
}
```

```
/*
 * 获取联系人
 *
 * @return
 */
private List<Map<String, Object>> getData() {
    // 组织数据源
    List<Map<String, Object>> list = new ArrayList<Map<String, Object>>();
    ContentResolver contentResolver = getContentResolver();
    Cursor cursor = contentResolver.query(
        ContactsContract.Contacts.CONTENT_URI, null, null, null, null);
    while (cursor.moveToNext()) {
        String id = cursor.getString(cursor
            .getColumnIndex(ContactsContract.Contacts._ID)); // id
        String name = cursor.getString(cursor
            .getColumnIndex(ContactsContract.Contacts.DISPLAY_NAME));

        Map<String, Object> map = new HashMap<String, Object>();
        map.put(CONTACT_ID, id);
        map.put(CONTACT_NAME, name);
        map.put(CONTACT_FACE, getContactBitmap(id));
        list.add(map);
    }
    cursor.close(); // 关闭
    return list;
}

/*
 * 获得联系人头像
 *
 * @param id
 *         联系人id
 * @return
 */
private Bitmap getContactBitmap(String id) {
    ContentResolver cr = getContentResolver();
    Uri uri = ContentUris.withAppendedId(
```

```
        ContactsContract.Contacts.CONTENT_URI, Long.parseLong(id));

    InputStream input = ContactsContract.Contacts
        .openContactPhotoInputStream(cr, uri);

    if (input == null) {
        // Log.e("star","input is null");

        InputStream is = getResources().openRawResource(R.drawable.icon);

        BitmapDrawable bmpDraw = null;
        bmpDraw = new BitmapDrawable(is);

        Bitmap aa = null;
        aa = bmpDraw.getBitmap();

        Bitmap newb = Bitmap.createScaledBitmap(aa, 50, 50, false);

        return newb;
    } else {
        Bitmap contactPhoto = BitmapFactory.decodeStream(input);

        Bitmap newb = Bitmap
            .createScaledBitmap(contactPhoto, 50, 50, false);

        return newb;
    }
}
```

}

ContactCheckBoxAdapter.java类:

```
package com.iaiai.activity.adapter;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

import android.content.Context;
import android.graphics.Bitmap;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
```

```
import android.widget.CheckBox;
import android.widget.ImageView;
import android.widget.TextView;

import com.iaiai.activity.ContactSelectorWidget;
import com.iaiai.activity.R;

public class ContactCheckBoxAdapter extends BaseAdapter {

    private LayoutInflator mInflater;

    private List<Map<String, Object>> listData; // 数据

    private Map<Integer, Map<String, Object>> selectMap = new HashMap<Integer, Map<String, Object>>();

    private class ViewHolder {
        public ImageView img;
        public TextView title;
        public CheckBox checkBox;
    }

    public ContactCheckBoxAdapter(Context context,
                                  List<Map<String, Object>> listData) {
        this.mInflater = LayoutInflator.from(context);
        this.listData = listData;
    }

    @Override
    public int getCount() {
        return listData.size();
    }

    @Override
    public Object getItem(int position) {
        return listData.get(position);
    }
}
```

```
@Override
public long getItemId(int position) {
    return position;
}

/**
 * 返回选择的数据 , Integer为选择的第几个
 *
 * @return
 */
public Map<Integer, Map<String, Object>> onSelectorData() {
    return selectMap;
}

@Override
public View getView(final int position, View convertView, ViewGroup parent) {
    ViewHolder holder = null;
    if (convertView == null) {
        holder = new ViewHolder();
        convertView = mInflater.inflate(R.layout.contact_list, null);
        holder.img = (ImageView) convertView.findViewById(R.id.contact_img);
        holder.title = (TextView) convertView
                .findViewById(R.id.contact_name);
        holder.checkBox = (CheckBox) convertView
                .findViewById(R.id.contact_checkbox);
        holder.checkBox.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                if (selectMap.get(position) != null) {
                    selectMap.remove(position);
                } else {
                    selectMap.put(position, listData.get(position));
                }
            }
        });
        convertView.setTag(holder);
    } else {

```

```
        holder = (ViewHolder) convertView.getTag();
    }

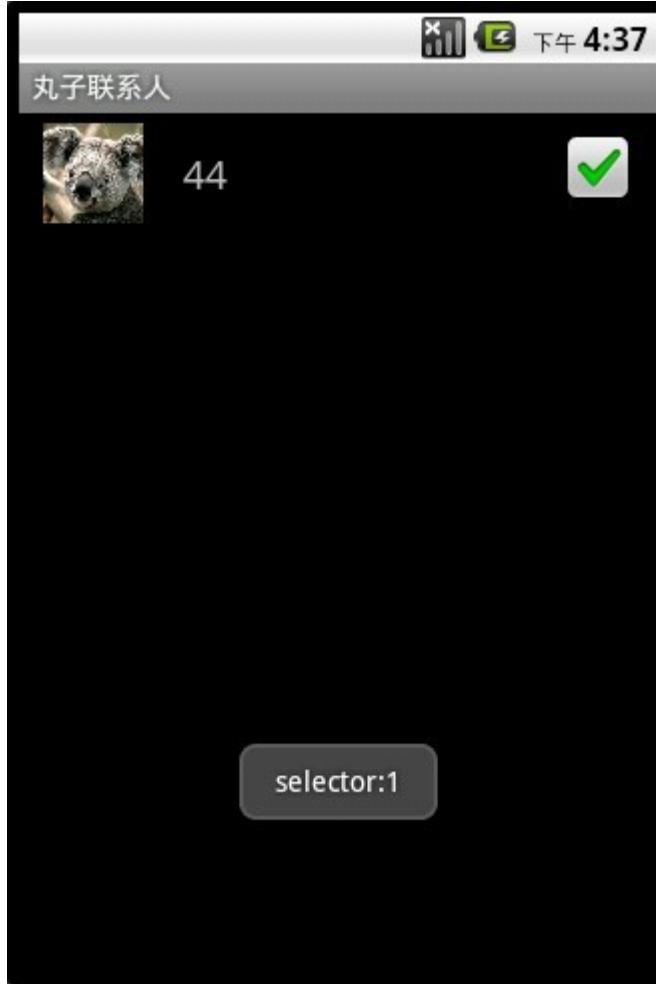
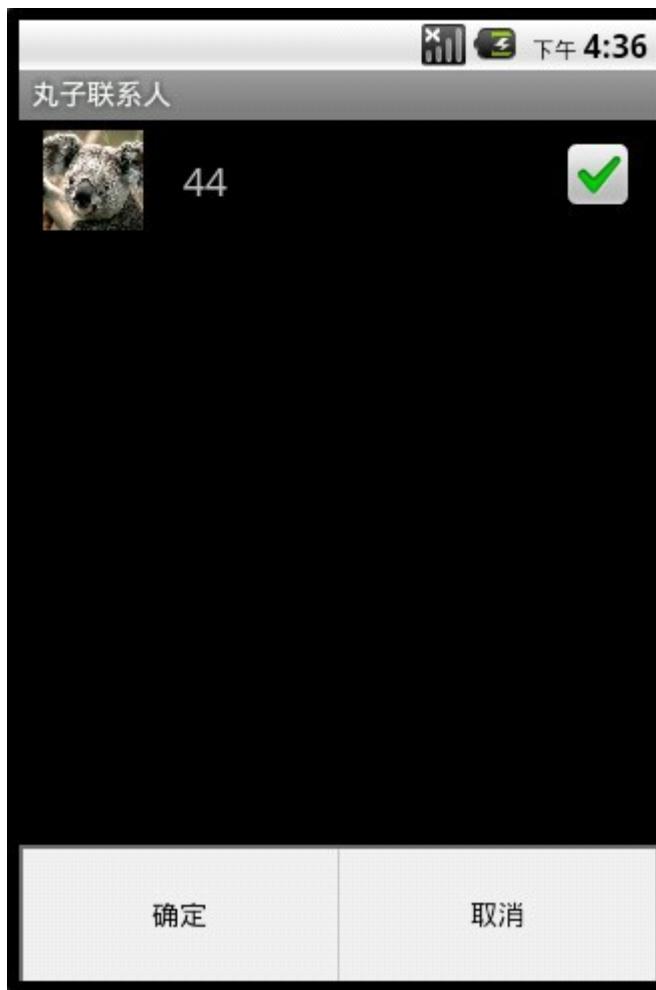
    holder.img.setImageBitmap((Bitmap) listData.get(position).get(
        ContactSelectorWidget.CONTACT_FACE));
    holder.title.setText(listData.get(position)
        .get(ContactSelectorWidget.CONTACT_NAME).toString());

    if (selectMap.get(position) != null) {
        holder.checkBox.setChecked(true);
    } else {
        holder.checkBox.setChecked(false);
    }

    return convertView;
}

}
```

运行结果:



2.2 Android QQ登录界面

发表时间: 2011-06-22 关键字: QQ, Android, XML, Java, ITeye

布局loginpage.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<LinearLayout android:orientation="vertical" android:id="@+id/loginRoot"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <LinearLayout android:orientation="vertical" android:id="@+id/LinearLayout01"
        android:layout_width="fill_parent" android:layout_height="wrap_content"
        android:layout_weight="1.0"
        android:background="@drawable/default_bg">
        <RelativeLayout android:id="@+id/RelativeLayout02"
            android:background="@drawable/login_back" android:paddingTop="21.0px"
            android:paddingBottom="10.0px" android:layout_width="fill_parent"
            android:layout_height="wrap_content" android:layout_marginLeft="15.0px"
            android:layout_marginTop="62.0px" android:layout_marginRight="15.0px">
            <ImageView android:id="@+id/faceImg" android:background="@drawable/login"
                android:layout_width="wrap_content" android:layout_height="wrap_content" />
            <EditText android:id="@+id/login_edit_account"
                android:background="@drawable/qq_edit_login" android:paddingLeft="10.0px"
                android:saveEnabled="true" android:layout_width="fill_parent"
                android:layout_height="wrap_content" android:layout_marginLeft="15.0px"
                android:layout_marginTop="5.0dip" android:layout_marginRight="15.0px"
                android:layout_marginBottom="5.0px" android:hint="@string/strAccInputLabel"
                android:maxLength="10" android:layout_toRightOf="@+id/faceImg"
                android:layout_alignParentTop="true" android:inputType="number" />
            <TextView android:textSize="16.0dip" android:textColor="#ff3f3f3f"
                android:gravity="center_vertical" android:id="@+id/TextView01"
                android:paddingLeft="7.0sp" android:layout_width="wrap_content"
                android:layout_height="wrap_content" android:layout_marginRight="15.0px"
                android:text="@string/strAccInputLabel" android:layout_alignLeft="@+id/login_edit_account"
                android:layout_alignTop="@+id/login_edit_account"
                android:layout_alignBottom="@+id/login_edit_account" />
            <ImageButton android:id="@+id/ImageButton02"
                android:background="@drawable/more_select" android:layout_width="wrap_content"
                android:layout_height="wrap_content" android:layout_alignRight="@+id/TextView01" />
        </RelativeLayout>
    </LinearLayout>
</LinearLayout>
```



```
        android:text="@string/opt_hideLogin" style="@style/MyCheckStyle" />
    <CheckBox android:textColor="#7fffffff" android:layout_gravity="right" android:id="@+id/login_vibration" android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="@string/opt_openVibra" style="@style/MyCheckStyle" />
</TableRow>
<TableRow>
    <CheckBox android:textColor="#7fffffff" android:id="@+id/login_silent" android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="@string/opt_acceptGroupMsg" style="@style/MyCheckStyle" />
    <CheckBox android:textColor="#7fffffff" android:layout_gravity="right" android:id="@+id/login_silent" android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="@string/opt_silent" style="@style/MyCheckStyle" />
</TableRow>
</TableLayout>
</LinearLayout>
<RelativeLayout android:id="@+id/RelativeLayout01" android:background="@drawable/bottom" android:layout_width="fill_parent" android:layout_height="44.0dip">
    <ImageButton android:id="@+id/login_option" android:background="@drawable/option" android:clickable="true" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_marginLeft="13.0dip" android:layout_marginTop="2.0dip" android:layout_centerVertical="true" />
</RelativeLayout>
</LinearLayout>
```

strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, Date!</string>
    <string name="app_name">丸子-Widget</string>
    <string name="strInputAccHint">输入帐号</string>
```

```
<string name="strAccInputLabel">帐号</string>
<string name="strPswInputLabel">密码</string>
<string name="opt_remember">记住密码</string>
<string name="login">登 录</string>
<string name="opt_hideLogin">隐身登录</string>
<string name="opt_silent">静音登录</string>
<string name="opt_openVibra">开启振动</string>
<string name="opt_acceptGroupMsg">接收群消息</string>
</resources>
```

styles.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
    <style name="MyCheckBox" parent="@android:style/Widget.CompoundButton.CheckBox">
        <item name="android:textSize">16.0sp</item>
        <item name="android:textColor">@color/qq_checkbox_text_color</item>
        <item name="android:paddingLeft">28.0px</item>
        <item name="android:button">@drawable/qq_btn_check</item>
    </style>
</resources>
```

运行结果:



附件下载:

- testAndroid.rar (1.1 MB)
- dl.iteye.com/topics/download/a82d0676-aebb-3af3-b3a7-e57b39d4e6c3

3.1 Using default keymap: /system/usr/keychars/qwerty.kcm.bin

发表时间: 2011-07-13

在文本框输入的时候如果报这种错:

引用

No keyboard for id 0

Using default keymap: /system/usr/keychars/qwerty.kcm.bin

说明是声卡的问题，有可能是驱动没装好，反正电脑上的声音必须可用，否则模拟器调用声音会出现问题。

3.2 Error inflating class xxx.xxx.xxx

发表时间: 2011-08-08

引用

Error inflating class xxx.xxx.xxx

生这个问题应该是你sdk版本设置问题，尝试修改项目的<uses-sdk android:minSdkVersion="7"/> 这个值即可，试试吧！

3.3 Parcelable encountered IOException writing serializable object (name = xxx)

发表时间: 2011-08-19

```
java.lang.RuntimeException: Parcelable encountered IOException writing serializable object (name = xxx)
    at android.os.Parcel.writeSerializable(Parcel.java:1160)
    at android.os.Parcel.writeValue(Parcel.java:1114)
    at android.os.Parcel.writeMapInternal(Parcel.java:479)
    at android.os.Bundle.writeToParcel(Bundle.java:1552)
    at android.os.Parcel.writeBundle(Parcel.java:493)
    at android.content.Intent.writeToParcel(Intent.java:5438)
    at android.app.ActivityManagerProxy.startActivity(ActivityManagerNative.java:1288)
    at android.app.Instrumentation.execStartActivity(Instrumentation.java:1373)
    at android.app.Activity.startActivityForResult(Activity.java:2817)
    at android.app.Activity.startActivity(Activity.java:2923)
    at android.widget.AdapterView.performItemClick(AdapterView.java:284)
    at android.widget.ListView.performItemClick(ListView.java:3382)
    at android.widget.AbsListView$PerformClick.run(AbsListView.java:1696)
    at android.os.Handler.handleCallback(Handler.java:587)
    at android.os.Handler.dispatchMessage(Handler.java:92)
    at android.os.Looper.loop(Looper.java:123)
    at android.app.ActivityThread.main(ActivityThread.java:4627)
    at java.lang.reflect.Method.invokeNative(Native Method)
    at java.lang.reflect.Method.invoke(Method.java:521)
    at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:868)
    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:626)
    at dalvik.system.NativeStart.main(Native Method)

Caused by: java.io.NotSerializableException: org.mdx.activity.collaboration.CollaborationListAdapter
    at java.io.ObjectOutputStream.writeNewObject(ObjectOutputStream.java:1547)
    at java.io.ObjectOutputStream.writeObjectInternal(ObjectOutputStream.java:1859)
    at java.io.ObjectOutputStream.writeObject(ObjectOutputStream.java:1701)
    at java.io.ObjectOutputStream.writeObject(ObjectOutputStream.java:1665)
    at java.io.ObjectOutputStream.writeFieldValues(ObjectOutputStream.java:1153)
    at java.io.ObjectOutputStream.defaultWriteObject(ObjectOutputStream.java:420)
    at java.io.ObjectOutputStream.writeHierarchy(ObjectOutputStream.java:1251)
    at java.io.ObjectOutputStream.writeNewObject(ObjectOutputStream.java:1587)
```

```
at java.io.ObjectOutputStream.writeObjectInternal(ObjectOutputStream.java:1859)
at java.io.ObjectOutputStream.writeObject(ObjectOutputStream.java:1701)
at java.io.ObjectOutputStream.writeObject(ObjectOutputStream.java:1665)
at android.os.Parcel.writeSerializable(Parcel.java:1155)
```

activity之间传递自定义数据，需要implements Serializable或者Parcelable接口

可以用bundle putSerializable (String , Serizlizable) 数据或者直接用intent putExtrr (String , Serizlizable) 数据。

我有一个需求需要传递多个自定义类型数据，于是我写了一个数据类Value把我需要的那些自定义数据封装了起来，这个类实现了Serizlizable接口。但是在putExtra的时候出现了错误。

后来我发现要想在Activity之间传递数据必须所有的内容都实现Serializable接口才行，而我的Value中封装的一个自定义类型数据没有实现这个接口，所以出现了错误。

之后我让那个自定义类型实现了Serializable接口就没有问题了。

4.1 android NDK下载地址

发表时间: 2011-08-06

android NDK下载地址

<http://developer.android.com/sdk/ndk/index.html>

4.2 Android API

发表时间: 2011-08-09

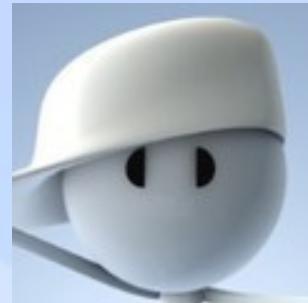
Android API

<http://developer.android.com/reference/packages.html>

丸子博客android文章精选

生成日期:2011-8-23

QQ:176291935



Android总结

作者: iaiai

<http://iaiai.iteye.com>

本书由ITEye提供电子书DIY功能制作并发行。

更多精彩博客电子书，请访问：<http://www.iteye.com/blogs/pdf>