



USB Camera 热插拔检测 接口说明

V1.1

2014-06-19

Revision History

Version	Date	Changes compared to previous issue
V1.0	2013-09-29	初始版本（By Microphone）
V1.1	2014-06-19	添加不导入 jar 包的使用方法

CONFIDENTIAL



目录

1. 接口说明.....	4
2. 接口的使用.....	5
3. 把源码中的 API 导入到 Eclipse 工程.....	6
4. 不导入 jar 包的使用方法.....	9

CONFIDENTIAL

1. 接口说明

接口的类：android.hardware.usb.UsbCameraManager

Constants		
String	UsbCameraManager.ACTION_USB_CAMERA_PLUG_IN_OUT	Broadcast Action

功能：上层通过监听与UsbCameraManager.ACTION_USB_CAMERA_PLUG_IN_OUT匹配的广播即可收到usb camera插拔的信息。

广播接收器回调函数中可以得到的信息：

接收匹配字段	获得的信息类型	说明
UsbCameraManager.USB_CAMERA_NAME	String	USB Camera设备节点的名称
UsbCameraManager.USB_CAMERA_STATE	int	插拔状态：1为插入，0为拔出
UsbCameraManager.USB_CAMERA_TOTAL_NUMBER	int	总共插入的摄像头个数
UsbCameraManager.EXTRA_MNG	String	附加描述信息

使用时需要：

```
import android.hardware.usb.UsbCameraManager;
```

2. 接口的使用

这里只做重点的说明。

- 1) 加入UsbCameraManager包

```
import android.hardware.usb.UsbCameraManager;
```

- 2) 把UsbCameraManager.ACTION_USB_CAMERA_PLUG_IN_OUT加到Intent过滤器中:

```
UsbCameraFilter = new IntentFilter();  
UsbCameraFilter.addAction(UsbCameraManager.ACTION_USB_CAMERA_PLUG_IN_OUT);
```

- 3) 注册广播接收器:

```
registerReceiver(UsbCameraBroadCastReceiver,UsbCameraFilter);
```

- 4) 接收器

```
private BroadcastReceiver UsbCameraBroadCastReceiver = new BroadcastReceiver() {  
  
    public void onReceive(Context context, Intent intent) {  
        // TODO Auto-generated method stub  
        String action = intent.getAction();  
        Log.d(TAG, "UsbCameraBroadCastReceiver Receive action: " + action);  
        if (action.equals(UsbCameraManager.ACTION_USB_CAMERA_PLUG_IN_OUT)){  
            Bundle bundle = intent.getExtras();  
            if(bundle == null){  
                Log.i(TAG,"bundle is null");  
                return;  
            }  
            //获取有用的信息  
            final String name = bundle.getString(UsbCameraManager.USB_CAMERA_NAME);  
            final int state = bundle.getInt(UsbCameraManager.USB_CAMERA_STATE);  
            final int totalNum =  
                bundle.getInt(UsbCameraManager.USB_CAMERA_TOTAL_NUMBER);  
            Log.i(TAG,"On UsbCamera device plug in/out receive,name=" + name + " totalNum=" +  
                totalNum + " state=" + state);  
        }  
    }  
};
```

- 5) AndroidManifest.xml中加入以下接收权限:

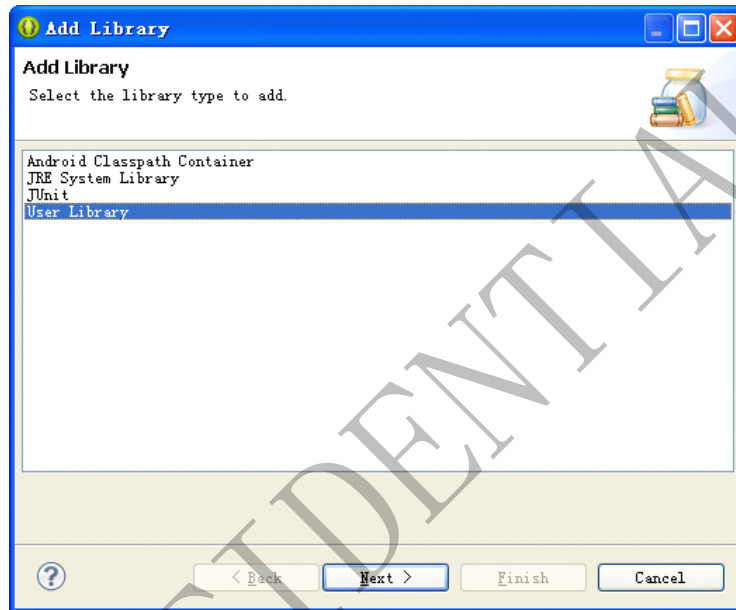
```
<uses-permission android:name="android.permission.BROADCAST_STICKY" />
```

3. 把源码中的 API 导入到 Eclipse 工程

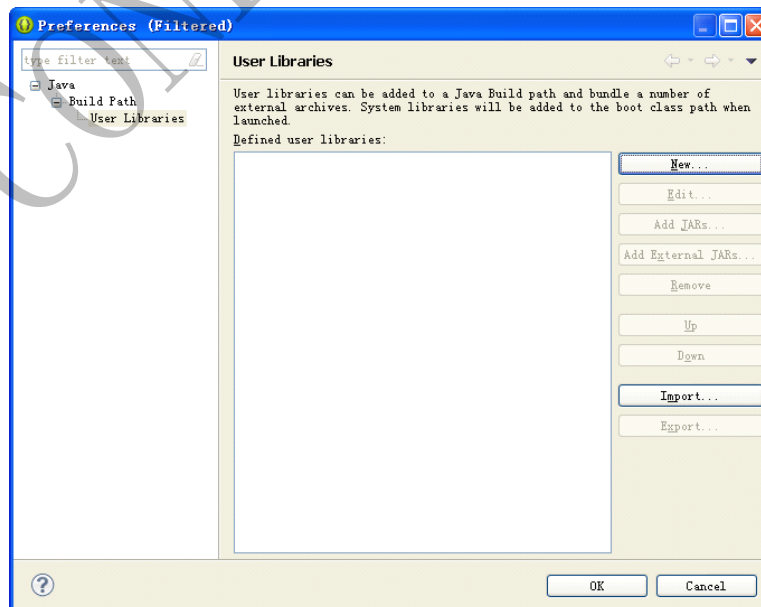
1) 源码的 API 文件为:

android4.2\out\target\common\obj\JAVA_LIBRARIES\framework_intermediates\classes.jar

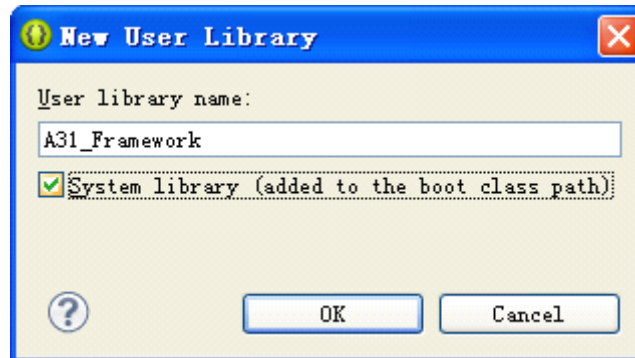
2) 右击该 android project,弹出菜单选 Build Path --> Add Libraries... --> User Library



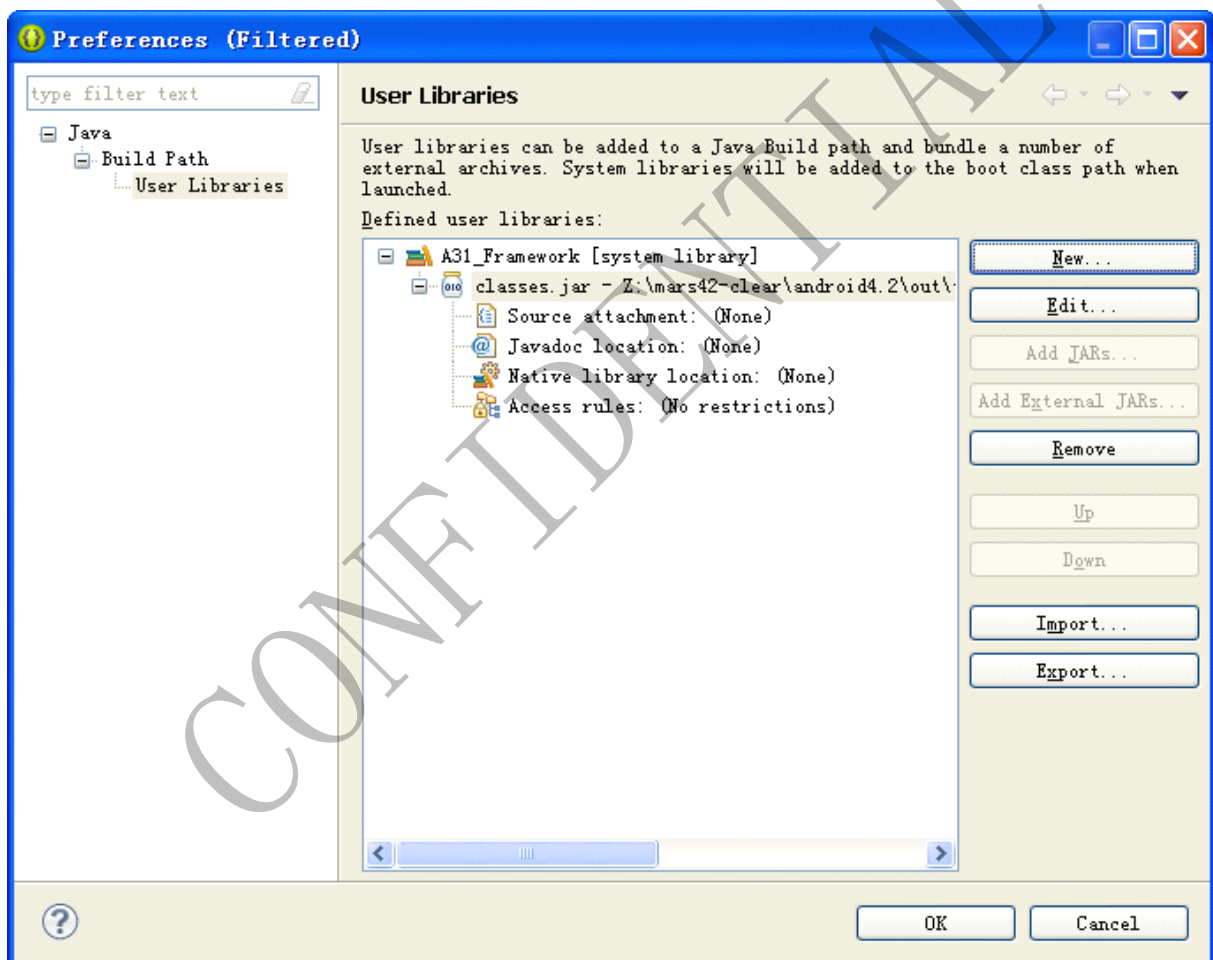
选择 next, 进入下图的界面:



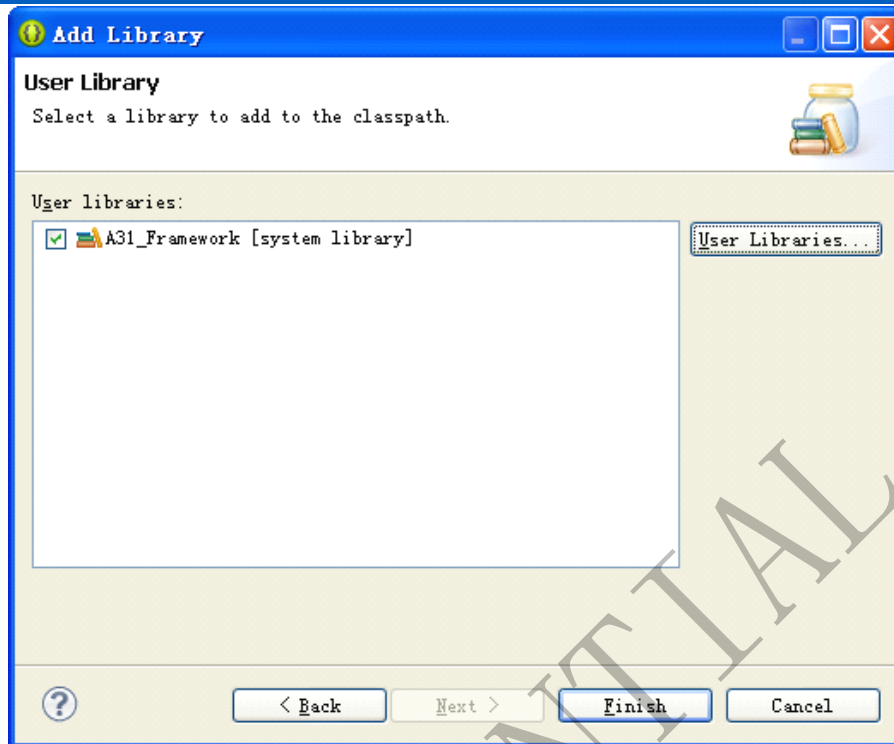
选择 new, 进入下图的界面:



如上图，输入名称后选择 OK，然后选择 Add external JARs，选择要导入 classes.jar 的路径，然后 OK 返回以下界面：

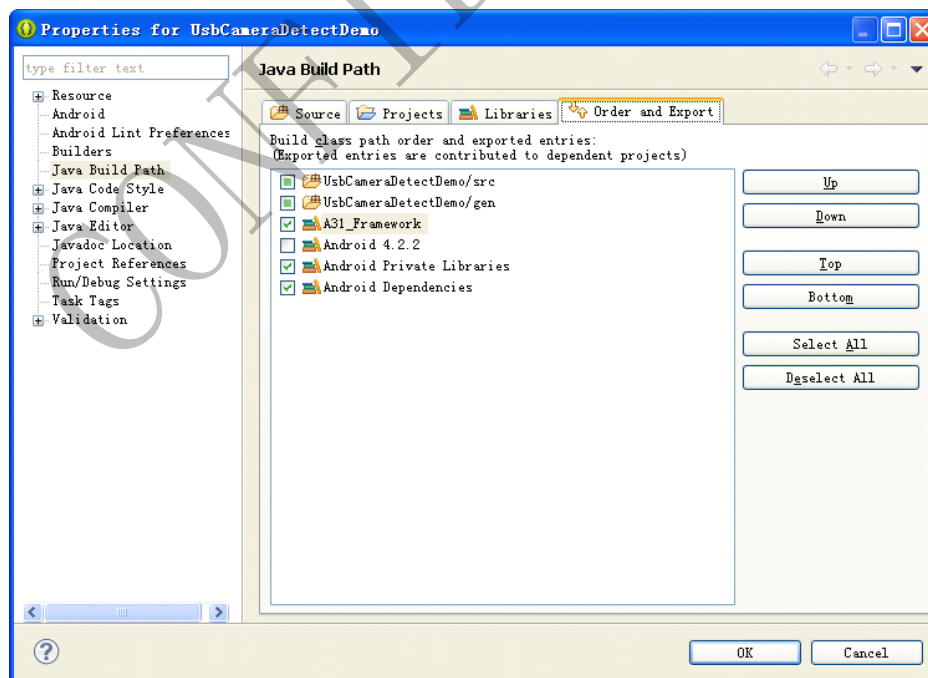


选择 OK，进入以下界面：



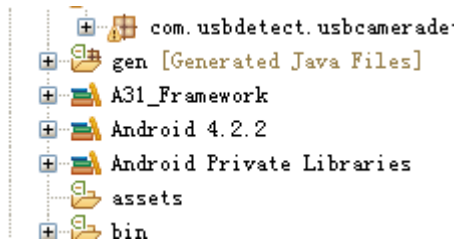
选择 Finish.

- 3) 右击该 project--> Build Path --> Configure Build Path --> Order and Export,把之前添加的 A31_framework 库 up 移到原生 SDK 的前面,编译时优先使用自己新增的库.如图:



选择 OK，完成添加。

4) 成功添加则可以在工程中看到:



4. 不导入 jar 包的使用方法

把 APK 中的监听关键字用以下的字符串代替接口。

```
public static final String USB_CAMERA_TOTAL_NUMBER = "UsbCameraTotalNumber";
public static final String USB_CAMERA_STATE = "UsbCameraState";
public static final int PLUG_IN = 1;
public static final int PLUG_OUT = 0;
public static final String USB_CAMERA_NAME = "UsbCameraName";
public static final String EXTRA_MNG = "extral_mng";

public static final String ACTION_USB_CAMERA_PLUG_IN_OUT =
    "android.hardware.usb.action.USB_CAMERA_PLUG_IN_OUT";
```

例如:

```
UsbCameraFilter.addAction(UsbCameraManager.ACTION_USB_CAMERA_PLUG_IN_OUT);
```

可以替换成:

```
UsbCameraFilter.addAction("android.hardware.usb.action.USB_CAMERA_PLUG_IN_OUT");
```

或者把上面的定义加到 APK 中, 直接已用。

```
UsbCameraFilter.addAction(ACTION_USB_CAMERA_PLUG_IN_OUT);
```

其他的字符也做相应的替代。这样就无需 import UsbCameraManage 包了。