



# 锐动 Android 手机录屏 SDK



作者：RD Android Team

版本：Beta 版

日期：2016-12-30

# 1 编写目的

预期读者：

有视频开发经验或者无经验的，打算或者正在使用“锐动 Android 录屏 SDK”的相关工程师。

- Android 软件工程师。
- 产品经理。
- QA

# 2 名词解释

- 分辨率：用于计算机视频处理的图像，以水平和垂直方向上所能显示的像素数来表示分辨率。常见视频分辨率的有 1080P 即 1920x1080，720P 即 1080x720，640x480 等。
- 帧率：每秒的帧数(fps)或者说帧率表示图形处理器处理场时每秒钟能够更新的次数。
- 码率：数据传输时单位时间传送的数据位数,一般我们用的单位是 kbps 即千位每秒。

# 3 集成步骤

## 3.1 运行环境

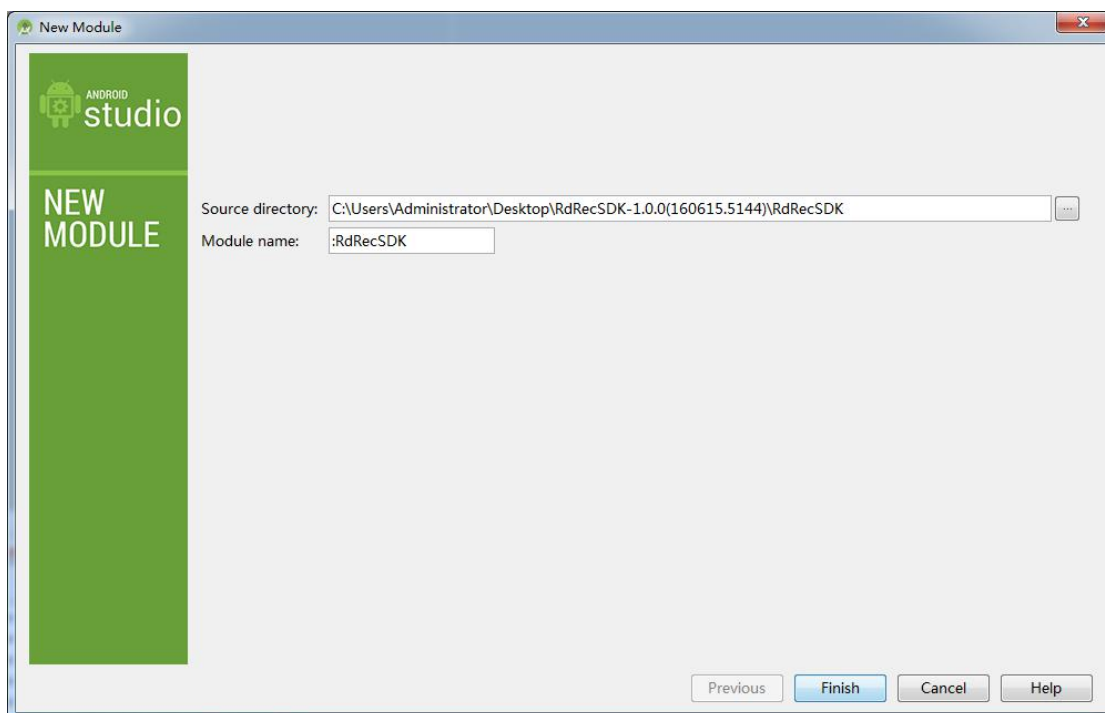
- Android 4.1（api 16+）以上；
- 处理器：双核 1GHz 以上 CPU(目前只支持 ARM CPU, X86、MIPS 暂不支持)；  
推荐四核 1.2GHz 以上 CPU

- 内存：1 GB 以上；

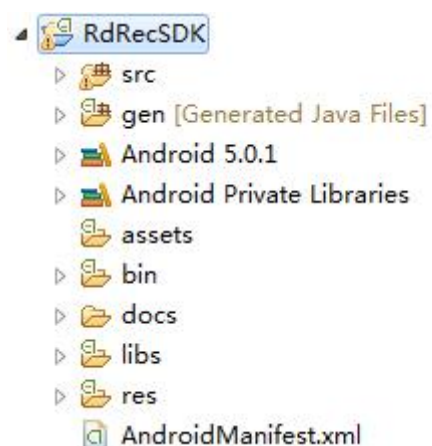
## 3.2 下载并导入 SDK

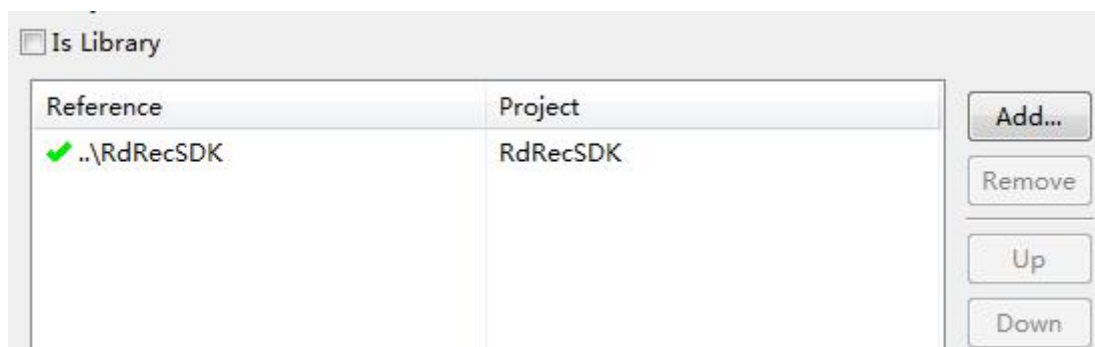
### 3.2.1 Android Studio 导入 RdRecSDK

1. 点击 File--->Import Module，选择路径，填写“Module name”，默认为 RdRecSDK，点击“finish”。



### 3.2.2 Eclipse 导入 RdRecSDK





## \*关于 arm 版本引发的冲突解决办法如下:

步骤一: sdk library 模块新增过滤

```
android {
    compileSdkVersion 19
    buildToolsVersion "23.0.2"
    packagingOptions {
        //过滤掉本地libs/armeabi和libs/armeabi-v7a中的部分so，以jcenter
        exclude 'lib/armeabi/libRdBase.so'
        exclude 'lib/armeabi/libLiveRecorder.so'
        exclude 'lib/armeabi/libRecorderKernel.so'
        exclude 'lib/armeabi-v7a/libRdBase.so'
        exclude 'lib/armeabi-v7a/libLiveRecorder.so'
        exclude 'lib/armeabi-v7a/libRecorderKernel.so'
    }
}
```

步骤二: application 模块中过滤

```
defaultConfig {
    minSdkVersion 16
    targetSdkVersion 23
    ndk {
        abiFilters "armeabi", "armeabi-v7a"
    }
}
```

### 3.2.3 准备 AndroidManifest.xml (权限, 注册 Activity)

a. 添加权限:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_SUPERUSER" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

## b.注册必要的 Activity 和 Service:

```
<!-- 录屏服务 -->
<service
    android:name="com.rd.recsdk.recsdkservice"
    android:process=":recsdk_service" />

<activity
    android:name="com.rd.sc.base.activity.VerifyActivity"
    android:theme="@android:style/Theme.Dialog" />
```

如果不想影响当前项目的 AndroidManifest.xml，可以合并 [RdRecSDK](#) 中的配置，修改 project.properties 文件如下：

```
android.library.reference.1=../RdRecSDK
```

```
manifestmerger.enabled=true
```

### 3.2.4 调用 sdk 初始化

- 1、登录 <http://www.rdsdk.com> 注册用户
- 2、登录注册好的用户
- 3、进入视频云管理 点击（新增）获取应用的 APPKEY 、APPSECRET

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    RecSdkManager.initilize(this, APPKEY, APPSECRET, true);
}
```

## 3.3 录屏

### 3.3.1 开始录屏

```
com.rd.recsdk.RecSdkManager.startRec(Context context, RecSdkConfig
recConfig, RecSdkMessageListener listener)
```

Parameters:

context 上下文

recConfig 录屏配置

recConfig.setSavePath(string str) //str 可传（本地路径、rtmp 路径、直播 Uid）

listener 回调消息 listener

```
public static interface RecSdkMessageListener {  
    /**  
     * 响应请求root权限失败  
     */  
    void onRequestRootFail();  
    /**  
     * 响应录制结束成功  
     */  
    void onRecordSuccess(String strRecordFile);  
    /**  
     * * 响应录制失败  
     * @param nResult  
     *      错误号<br>  
     *      参考定义:RecSdkManage.RESULT*,如通用错误  
     *      {@link RecSdkManager#RESULT_ERROR}<br>  
     *      {@link RecSdkManager#RESULT_ERROR_DUPLICATE_RECORD}<br>  
     *      {@link RecSdkManager#RESULT_ERROR_CANNOT_PAUSE}<br>  
     *      {@link RecSdkManager#RESULT_ERROR_CANNOT_CONTINUE}<br>  
     *      {@link RecSdkManager#RESULT_NOAUTH}<br>  
     *      {@link RecSdkManager#RESULT_ERROR_FOR_UID_LIVE}<br>  
     *      {@link RecSdkManager#RESULT_ERROR_FOR_SIGN}<br>  
     *      {@link RecSdkManager#RESULT_ERROR_FOR_NET}<br>  
     * @param msg  
     *      失败原因说明<br>  
     */  
    void onRecordFailed(int nResult, String msg);  
    /**  
     * 响应开始录制  
     */  
    void onRecordBegin();  
}
```

### 3.3.2 暂停\继续录屏

void com.rd.recsdk.RecSdkManager.pauseRec(Context context)

Parameters:

context 上下文

void com.rd.recsdk.RecSdkManager.continueRec(Context context)

Parameters:

context 上下文

### 3.3.3 结束录屏

```
void com.rd.recsdk.RecSdkManager.endRec(Context context)
```

Parameters:

`context` 上下文

### 3.3.4 请求并检查 root 权限

```
boolean com.rd.recsdk.RecSdkManager.requestRootPermission(Context  
context)
```

请求 root 权限，如果 api21+，可以不用请求

Parameters:

`context` 上下文

Returns:

返回 `true` 代表成功获取 root 权限，否则代表获取失败

## 3.4 录屏配置

### 3.4.1 录屏配置实体类

```
RecSdkConfig config = new RecSdkConfig();
```



### 3.4.2 具体配置信息

int	<code>getCapMode()</code> 获取录屏方式
int	<code>getRecEncodeMode()</code> 获取录制编码模式
int	<code>getRecVideoDefinition()</code> 获取视频质量定义
java.lang.String	<code>getSavePath()</code> 获取保存路径
boolean	<code>isRecAudio()</code> 获取是否录制声音
boolean	<code>isSwapColor()</code> 获取是否需要修正颜色
void	<code>loadFromSharedPreferences(android.content.Context context)</code> 读取持续化存储
void	<code>saveToSharedPreferences(android.content.Context context)</code> 持续化存储
void	<code>setCapMode(int capMode)</code> 设置录屏方式
void	<code>setDefaultConfig()</code> 设置默认配置
void	<code>setRecAudio(boolean recAudio)</code> 设置是否录制声音
void	<code>setRecEncodeMode(int encMode)</code> 设置录制编码模式
void	<code>setRecVideoDefinition(int videoDefinition)</code> 设置视频质量定义
void	<code>setSavePath(java.lang.String savePath)</code> 设置保存路径
void	<code>setSwapColor(boolean swapColor)</code> 设置是否修正颜色

通过 set/get 方法来修改/获取具体配置信息，需要持续化存储可以通过  
saveToSharedPreferences/loadFromSharedPreferences 方法进行保存与加载

## 3.5 直播权限

```
/**
 * 返回当前SDK支持的直播方式<br>
 * -1代表Uid与Url直播方式都不可用 {@link RecSdkManager#AT_INVALID}<br>
 * 2代表仅支持Uid直播方式 {@link RecSdkManager#AT_UID}<br>
 * 3代表仅支持Url直播方式 {@link RecSdkManager#AT_URL}<br>
 * 1代表两种直播方式都支持 {@link RecSdkManager#AT_URL_OR_UID}<br>
 *
 * @return 当前SDK支持的直播方式
 */
int aType = RecSdkManager.getAuthType();
```



## 3.6 看直播

注意：看 UID 直播需要获取网络数据（直播地址、liveId）

```
RecSdkManager.getLivingUid(uid,
    new RecSdkManager.ILivingListener() {

        @Override
        public void getLiving(
            HashMap<String, String> maps) {
            if (maps.containsKey("error")) {
                onToast(maps.get("error"));
            } else {
                String rtmp = maps
                    .get("rtmp");
                String m3u8 = maps
                    .get("m3u8");
                Log.d("getdata", "rtmp->"
                    + rtmp);
                Log.d("getdata", "m3u8->"
                    + m3u8);
                PlayActivity.gotoPlay(
                    MainActivity.this,
                    rtmp,
                    maps.get("thumb"),
                    maps.get("title"),
                    maps.get("liveId"));
            }
        }
    });
```

播放器参数:

```
vvPlay = (RdLivePlayer) findViewById(R.id.vv_play);  
vvPlay.setTimeout(PLAYER_TIMEOUT);  
flIndicator = findViewById(R.id.fl_indicator);  
judgeLive(playTag);  
vvPlay.setMediaBufferingIndicator(flIndicator);  
vvPlay.setVideoPath(liveUrl, liveId);  
vvPlay.requestFocus();
```

## 3.5 打包混淆

需要在 proguard.cfg 文件中添加如下配置项：

```
-dontwarn com.rd.**  
  
-keep class com.rd.** { *; }  
  
-dontwarn tv.danmaku.**  
  
-keep class tv.danmaku.** { *; }
```