

手机支付控件接入指南

3.2.0

中国银联

2018-11-19

版本信息

文档版本	日期	说明	适用 SDK 版本
2.0.0	2013-01-08	初稿	
2.0.1	2015-07-29	部分错误修正。 安卓升级 3.0.7 控件 data.bin 移到 assets 文件夹下。 增加 jar 集成方式说明。 删除浏览器调起控件相关说明，浏览器方式请直接使用 wap 产品。 增加 FAQ。	--
3.0.0	2015-12-09	调用接口统一为 startPay 增加 UPPayWapActivity 增加 libuptsmaddon.so	3.3.2 及以上
3.1.0	2017-11-24	增加检查手机 pay 状态的接口 增加指定手机 pay 支付接口 增加 libuptsmaddonmi.so	3.4.1 及以上
3.1.1	2018-1-12	增加手机 pay 入口调用说明与示例代码	3.4.1 及以上
3.1.2	2018-2-23	细化 4.2 接口使用场景与要求	3.4.1 及以上
3.2.0	2018-11-20	增加接口判断云闪付 APP 是否安装接口 去除原有的 checkInstalled 方法 更新 4.3 节 Proguard 规则	3.4.5 及以上

目录

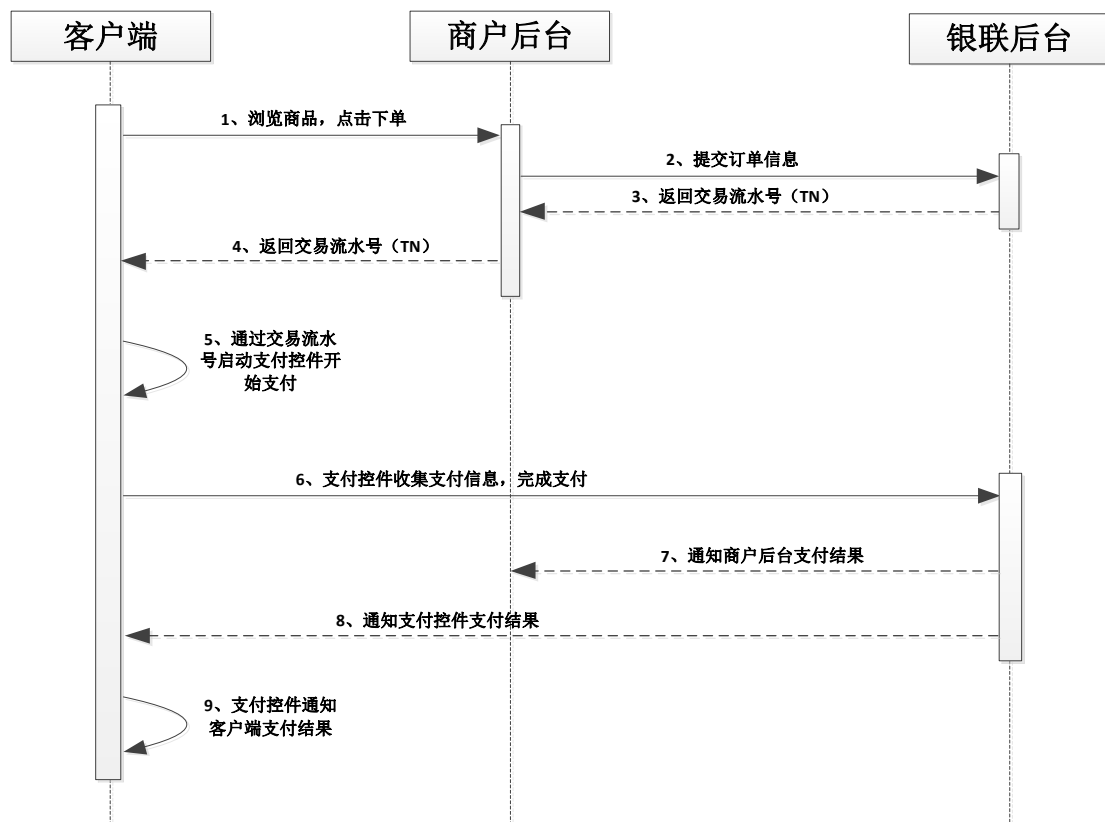
版本信息	2
目录	3
1 概述	1
2 支付流程介绍	1
3 测试帐号	2
4 Android 客户端	3
4.1 SDK 包说明	3
4.2 接口说明	4
4.3 添加 SDK 包	7
4.4 调用支付控件	10
5 老版 SDK 商户修改说明	13
6 常见问题	14

1 概述

银联手机支付控件(以下简称支付控件)，主要为合作商户的手机客户端提供安全、便捷的支付服务。用户通过在支付控件中输入银行卡卡号、手机号、密码（借记卡和预付卡）或者 CVN2、有效期（信用卡）、验证码等要素完成支付。

2 支付流程介绍

通过支付控件进行交易的流程如下图：



流程图说明：

- （1）用户在客户端中点击购买商品，客户端发起订单生成请求到商户后台；
- （2）商户后台收到订单生成请求后，按照《手机控件支付产品接口规范》组织并推送订单信息至银联后台；

- (3) 银联后台接收订单信息并检查通过后，生成对应交易流水号（即 TN），并回复交易流水号至商户后台（应答要素：交易流水号等）；
- (4) 商户后台接收到交易流水号，将交易流水号返回给客户端；
- (5) 客户端通过交易流水号（TN）调用支付控件；
- (6) 用户在支付控件中输入相关支付信息后，由支付控件向银联后台发起支付请求；
- (7) 支付成功后，银联后台将支付结果通知给商户后台；
- (8) 银联将支付结果通知支付控件；
- (9) 支付控件显示支付结果并将支付结果返回给客户端；

注：本文档主要关注上述流程中（5）、（9）部分的实现

目前各个平台支持的设备情况如下：

Android 平台 SDK 主要适用于 **Android 2.3** 及以上版本的终端设备；

3 测试帐号

提供测试使用卡号、手机号信息（**此类信息仅供测试，不会发生正式交易**）

招商银行借记卡：62260900000000048

手机号：18100000000

密码：111101

短信验证码：123456（**先点获取验证码之后再输入**）

证件类型：01 身份证

证件号：510265790128303

姓名：张三

华夏银行贷记卡：6226388000000095

手机号：18100000000

cvn2：248

有效期：1219

短信验证码：123456 (先点获取验证码之后再输入)

证件类型：01 身份证

证件号：510265790128303

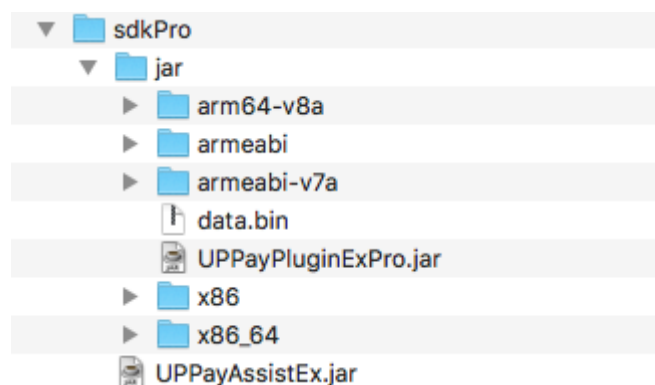
姓名：张三

4 Android 客户端

本小节涉及到 SDK 的接口说明、及其接口调用细节，需要读者具有一定 Android 编程经验。

4.1 SDK 包说明

商户开发者在获取到银联提供的开发包后请检查 SDK 文件所在目录 upmp_android/sdkPro，以下部分提及的文件均在该目录中：



jar 目录下包括了商户集成所需要的 jar 包、so 文件（支持 arm,armv7,x86 , x86_64 和 arm64-v8a 平台）和资源文件。

UPPayAssistEx.jar 定义了调用支付控件所需要的接口。

4.2 接口说明

对于商户收银页面，普通支付入口应始终显示并支持，手机 pay 入口应根据手机情况判断后显示支持。视觉要求参见开发包内“手机控件商户收银台品牌露出规范”



- 普通支付入口接入方式为：直接调用“普通支付接口”
- 手机 pay 支付入口接入方式为：首先商户通过“检查手机 pay 状态的接口”获取当前手机 pay 种类，收银页面展示；其次使用“指定手机 pay 支付接口”调用该手机 pay。

1、普通支付接口

upmp_android/UPPayAssistEx.jar中定义了启动支付控件的接口，接口定义如下：

```
public static int startPay(Activity activity, String spId, String sysProvider, String orderInfo, String mode)
```

参数说明：

activity — 用于启动支付控件的活动对象

spId — 保留使用，这里输入`null`
sysProvider — 保留使用，这里输入`null`
orderInfo — 订单信息为交易流水号，即TN，为商户后台从银联后台获取。
mode — 银联后台环境标识，“00”将在银联正式环境发起交易，“01”将在银联测试环境发起交易

2、检查手机 pay 状态的接口

本接口用于判断当前设备手机 pay 状态的支持情况。商户需本接口返回值，在收银页面动态展示对应的手机 pay 支付，对应图标参照开发包内的“商户收银台品牌露出规范”（持续更新），实现对表 1 所述的**全部手机 pay 的支持**。

```
public static int getSEPayinfo(Context context, UPQuerySEPay InfoCallback
callback)
```

参数说明：
context — 用于获取启动支付控件的活动对象的context
callback — 该接口定义如下：

```
public interface UPQuerySEPayInfoCallback {
    public void onResult(String SENAME, String seType, int cardNumbers,
        Bundle reserved);

    public void onError(String SENAME, String seType, String errorCode,
        String errorDesc);
}
```

需要实现两个方法，分别是

```
public void onResult(String SENAME, String seType, int cardNumbers, Bundle
reserved);
```

该方法在正常情况调用，参数如下：

- SEName** — 手机 pay 名称，如表 1
- seType** — 与手机 pay 名称对应的类别，如表 1
- cardNumbers** — 卡数量
- reserved** — 保留字段

```
public void onError(String SENAME, String seType, String errorCode, String
errorDesc);
```

该方法会在检测到异常时调用，参数如下

- SEName** — 手机 pay 名称，如表 1
- SeType** — 与手机 pay 名称对应的类别，如表 1
- errorCode** — 错误情况及对应的错误码：
 - 1.XXpay 未安装或 TSM 控件版本低 `ERROR_NOT_SUPPORT = "01"`
 - 2.硬件不支持 XXpay `ERROR_NOT_SUPPORT = "01"`

3. 未开通 XXpay `ERROR_NOT_READY = "02"`
4. 可用卡数为 0 `ERROR_NOT_READY = "02"`
5. 检测未安装 TSM 控件 `ERROR_NOT_SUPPORT = "04"`
6. 接口超时 `ERROR_TIMEOUT = "10"`

`errorDesc` — 根据错误码提示相应的字符串

返回值:

`UPSEInfoResp.SUCCESS` — 成功获取 SEPay 状态

`UPSEInfoResp.PARAM_ERROR` — 传入参数有一个以上为空, 获取失败

SEName与seType对应关系表格如下:

表1 SEName与seType对应表格

SEName	seType
Samsung Pay	02
Huawei Pay	04
Meizu Pay	27
Le Pay	30
ZTE Pay	21
Mi Pay	25
vivo Pay	33
Smartisan Pay	32

3、指定手机 pay 支付接口

```
public static int startSEPay(Context context, String spId, String sysProvider, String orderInfo, String mode, String seType)
```

参数说明:

`context` — 用于获取启动支付控件的活动对象的context

`spId` — 保留使用, 这里输入`null`

`sysProvider` — 保留使用, 这里输入`null`

`orderInfo` — 订单信息为交易流水号, 即TN, 为商户后台从银联后台获取。

`mode` — 银联后台环境标识, “00”将在银联正式环境发起交易, “01”将在银联测试环境发起交易

`seType` — 手机pay支付类别, 见表1

返回值:

0

4、检查是否安装云闪付客户端的接口

upmp_android/UPPayAssistEx.jar中定义了检测云闪付客户端是否安装的接口，

接口定义如下：

```
public static boolean checkWalletInstalled (Context context)
```

参数说明：

activity — 用于启动支付控件的context环境

返回值：

true — 该终端已经安装控件apk

false — 该终端未安装控件apk

4.3 添加 SDK 包

添加资源文件：

1. 拷贝 upmp_android/sdkPro/jar/data.bin 到工程的 assets/目录下；

2. 拷贝 upmp_android/sdkPro/jar/xxx/libentryexpro.so、

upmp_android/sdkPro/jar/xxx/libuptsmaddon.so 和

upmp_android/sdkPro/jar/xxx/libuptsmaddonmi.so 到工程的 libs/xxx/目录

下，其中 xxx 为 armeabi-v7a、armeabi、arm64-v8a、x86、x86_64 之一。

arm64-v8a(x86_64)是针对 arm64(x86 64)架构优化的库文件，引入工程后在

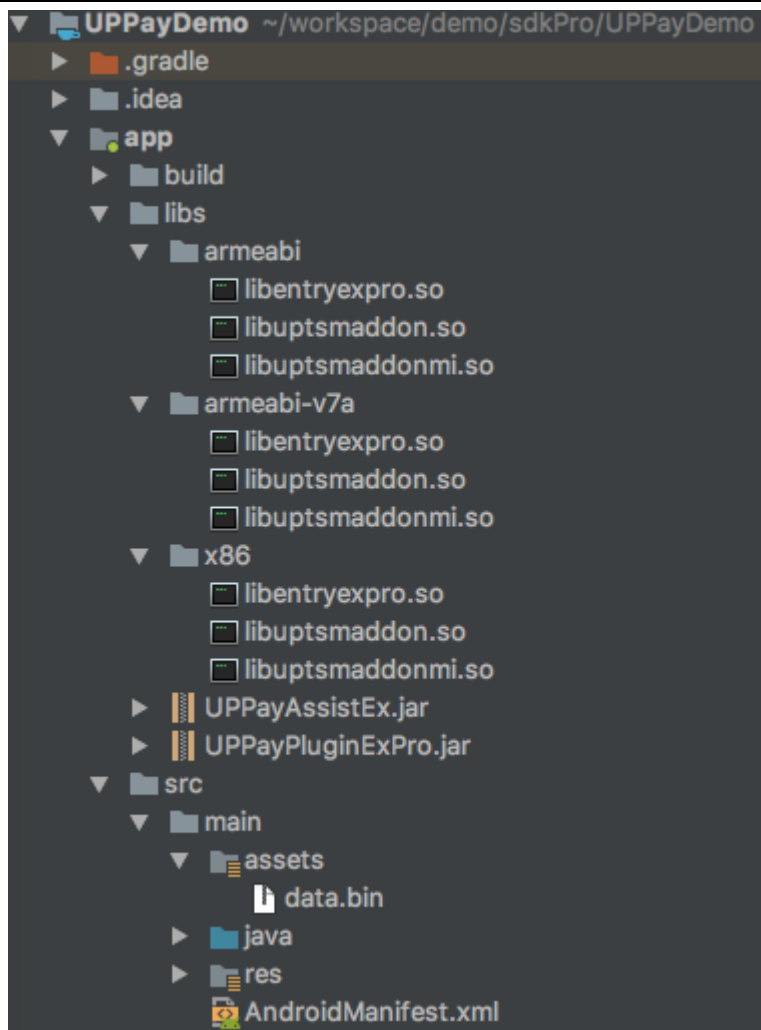
arm64(x86 64)机型上性能会提升，但是最终生成的程序包将变大。**注意：如果工**

程中使用了其他的.so 库，那么需要所有.so 库都有 arm64-v8a(x86_64)的版本。

3. 拷贝upmp_android/sdkPro/UPPayAssistEx.jar到工程的libs/目录下；

4. 拷贝upmp_android/sdkPro/jar/UPPayPluginExPro.jar到工程的libs/目录下；

效果如下图：



arm64-v8a (x86_64)是针对 arm64(x86_64)架构优化的库文件，引入工程后在 arm64(x86_64)机型上性能会提升，但是最终生成的程序包将变大。**注意：如果工程中使用了其他的.so 库，那么需要所有.so 库都有 arm64-v8a(x86_64)的版本。**

5. 在工程的AndroidManifest.xml文件中注册支付插件使用的Activity。添加如下：

```
<application>
<!--工程其它配置此处省略--->
<uses-library
    android:name="org.simalliance.openmobileapi"
    android:required="false"/>
<activity
    android:name="com.unionpay.uppay.PayActivity"
    android:label="@string/app_name"
```

```

        android:screenOrientation="portrait"
        android:configChanges="orientation|keyboardHidden"
        android:excludeFromRecents="true"
        android:windowSoftInputMode="adjustResize"/>

<activity
    android:name="com.unionpay.UPPayWapActivity"
    android:configChanges="orientation|keyboardHidden"
    android:screenOrientation="portrait"
    android:windowSoftInputMode="adjustResize"/>
</application>

```

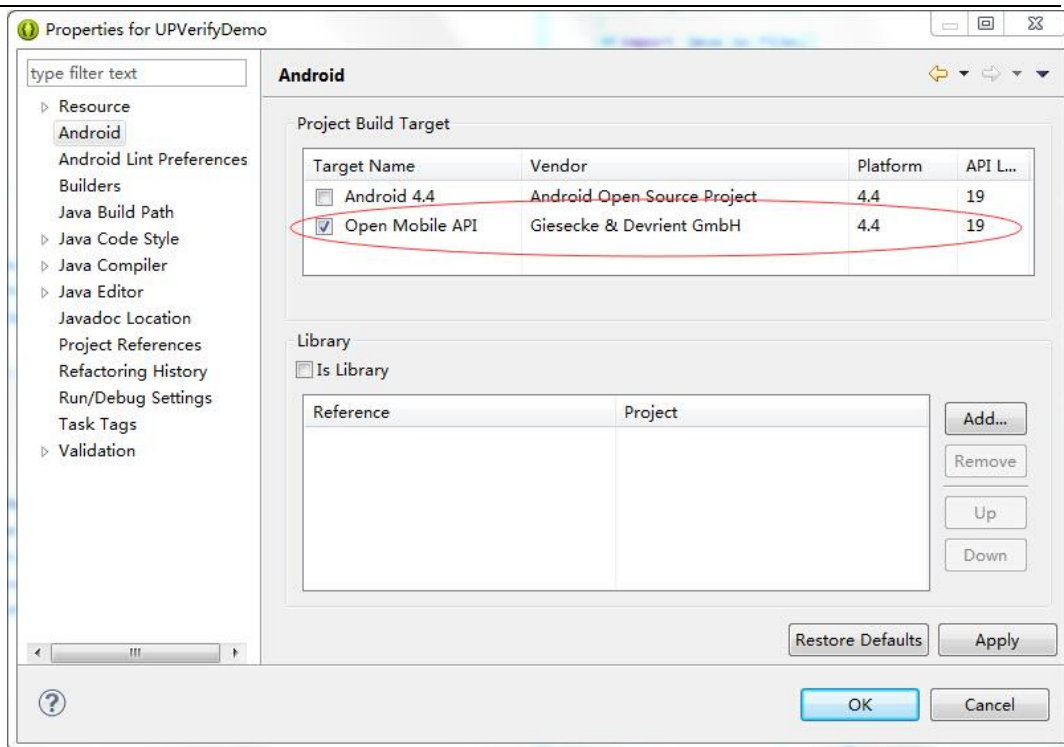
同时添加，银联支付插件相关权限：

```

<uses-permission
    android:name="android.permission.INTERNET"/>
<uses-permission
    android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission
    android:name="android.permission.CHANGE_NETWORK_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.NFC" />
<uses-feature android:name="android.hardware.nfc.hce"/>
<uses-permission android:name="org.simalliance.openmobileapi.SMARTCARD" />

```

注意：版本编译时的 target 应选用 Open Mobile API，建议 level19 以上。使用 eclipse 开发，如下图所示：



使用 Android Studio , 可将 Open Mobile API 的 jar 包放在 libs 文件夹中 , 添加如下
依赖配置 :

```
provided files('libs/org.simalliance.openmobileapi.jar')
```

6. Proguard规则

```
//在proguard文件中添加如下规则
-dontwarn com.unionpay.**
-keep class com.unionpay.** {*;}
-keep class org.simalliance.openmobileapi.** {*;}
```

注意：集成时，务必将上述Proguard规则加入

4.4 调用支付控件

1、支付接口调用

a) 在调用支付控件的代码文件中引入UPPayAssistEx类如：

```
import com.unionpay.UPPayAssistEx;
```

b) 接着可以通过以下方式调用支付控件：

```
// “00” - 银联正式环境
// “01” - 银联测试环境，该环境中不发生真实交易
```

```
String serverMode = "01";
UPPayAssistEx.startPay (activity, null, null, tn, serverMode);
```

c) 控件回调说明：

支付完成后，获取支付控件支付结果，并添加相应处理逻辑，只需实现调用Activity中的onActivityResult()方法即可，支付成功时会返回商户客户端支付结果和签名信息。

控件返回的结果信息仅作为参考，商户订单是否成功支付应该以商户后台主动到全渠道查询的结果或者收到全渠道支付结果通知为准。

收到控件返回结果后，无需检查结果中的内容和签名信息，请直接使用后台进行状态查询，以后台状态查询为准。如果后台此时未收到全渠道的后台通知，则主动发起查询接口到全渠道查状态。这样可既保证APP中展示的订单状态和后台记录一致，也可及时更新后台记录的订单状态。

如仍希望使用控件返回的信息，验签时建议送到后台去验签；不应在APP中验签，否则银联更新密钥后会验签失败。

示例代码如下：

```
protected void onActivityResult( int requestCode,
    int resultCode,
    Intent data)
{
    if( data == null ){
        return;
    }
    String msg = "";
    String str = data.getExtras().getString("pay_result");
    if( str.equalsIgnoreCase(R_SUCCESS) ){

        // 如果想对结果数据验签，可使用下面这段代码，
        // 但建议不验签，直接去商户后台查询交易结果
        // result_data结构见c) result_data参数说明
        if(data.hasExtra("result_data")) {
            String sign = data.getExtras().getString("result_data");
            // 此处的verify建议送去商户后台做验签
```

```

// 如要放在手机端验，则代码必须支持更新证书
if(verify(sign)) {
    //验签成功，显示支付结果
    showResultDialog(" 支付成功! ");
} else {
    // 验签失败
}

// 结果result_data为成功时，去商户后台查询一下再展示成功
}else if( str.equalsIgnoreCase(R_FAIL) ){
    showResultDialog(" 支付失败! ");
}else if( str.equalsIgnoreCase(R_CANCEL) ){
    showResultDialog(" 你已取消了本次订单的支付! ");
}
}

```

result_data参数说明：

参数说明：

sign — 签名后做Base64的数据

data — 用于签名的原始数据

data中原始数据结构：

pay_result — 支付结果success, fail, cancel

tn — 订单号

result_data示例如下：

```

{ "sign" :
"Xo/pgkzSJS1RTX2e+CjW/k1IjIV1newqfb7p1sDIpK/yPQv9p1jQAdAdKwhBwtyj03tkFC6I2a
LcTaxLH1YQx6/xw9QE0eumkVqAhypk/VyoDWZXxWske+EcduwEkBTxyIgA0ZsbKlpS1Jxsci0c6
bT+f36jTLa05ZAKZTVerg9sAG3wMjae1TyKd2511Rvvi+tuihYgOmwuMnKzrqksEyqc69wloqi3
4qx0YqFo1MeqQ1UfoglUhZy6s2s4ChKcxHjAFjp/rU/7iHudjAIGt07+ySahArmw6ltuIxFWYEv
pn5xI3Ceur1d11NBphK62it7kBZ1laxUFI98DzalVFQ==", "data" :
"pay_result=success&tn=899394085660622736701&cert_id=68759585097"}

```

2、检测是否已安装云闪付客户端接口调用

```

if(UPPayAssistEx.checkWalletInstalled(context)) {
    //当判断用户手机上已安装银联Apk，商户客户端可以做相应个性化处理
}

```

3、检查手机 Pay 状态接口调用

```
UPQuerySEPayInfoCallback callback = new UPQuerySEPayInfoCallback() {

    @Override
    public void onResult(String SENAME, String seType, int cardNumbers,
        Bundle reserved) {
        // 该方法在手机Pay正常情况下回调
    }

    @Override
    public void onError(String SENAME, String seType, String errorCode,
        String errorDesc) {
        // 该方法在手机Pay异常情况下回调
    }
};
int ret = UPPayAssistEx.getSEPayInfo(activity, callback);
```

4、指定手机 Pay 支付接口调用

调用指定手机Pay支付接口 (startSEPay()) 之前，需要先调用检查手机Pay状态接口 (getSEPayInfo()) 获取seType，startSEPay()调用方式同startPay()。

5 老版 SDK 商户修改说明

对于原有sdk接入商户，可在原有修改的基础上更新对应文件后，还需增加如下修改：

1. 增 加 upmp_android/sdkPro/jar/xxx/libuptsmaddon.so 和
upmp_android/sdkPro/jar/xxx/libuptsmaddonmi.so
2. 在工程的AndroidManifest.xml文件增加UPPayWapActivity。添加如下：

```
<application>
```



```
<!--工程其它配置此处省略...-->

<activity
    android:name="com.unionpay.UPPayWapActivity"
    android:configChanges="orientation|keyboardHidden"
    android:screenOrientation="portrait"
    android:windowSoftInputMode="adjustResize"/>

</application>
```

3. 调用接口改为4.2节中支付接口startPay，但原有的startPayByJAR接口仍保留，如下所示：

```
public static int startPayByJAR (Activity activity, Class<?> payCls,
String spId, String sysProvider, String orderInfo, String mode)
```

参数说明：

activity — 用于启动支付控件的活动对象

payCls — 支付插件代表的类，填入“PayActivty.**class**”即可

spId — 保留使用，这里输入**null**

sysProvider — 保留使用，这里输入**null**

orderInfo — 订单信息为交易流水号，即TN，为商户后台从银联后台获取。

mode — 银联后台环境标识，“00”将在银联正式环境发起交易，“01”将在银联测试环境发起交易

6 常见问题

请参见 <https://open.unionpay.com> 帮助中心-FAQ