手机支付控件接入指南

3.0.0

中国银联

2015-12-09

版本信息

版本号	日期	说明
2.0.0	2013-01-08	初稿
2.1.0	2015-09-16	1、修正部分错误 2、删除浏览器调起控件相关说明，浏览器方式请参考web支付相关
3.0.0	2015-12-09	1、使用UPPaymentControl替换老版本支付控件 2、修改了接口说明和调用插件说明 3、增加了4.4工程配置说明 4、增加了老版本银联商户修改说明

目录

版本信息.....2

1. 概述 1

2. 支付流程介绍 1

3. 测试帐号 2

4. iOS客户端 2

4.1. SDK说明 3

4.2. 接口说明 3

4.3. 添加SDK包 5

4.4. 工程配置 5

4.5. 调用插件 8

5. 老版控件用户修改说明 10

5.1. 添加SDK包 10

5.2. 工程配置 11

5.3. 接口说明和调用插件 13

6. 常见问题 14

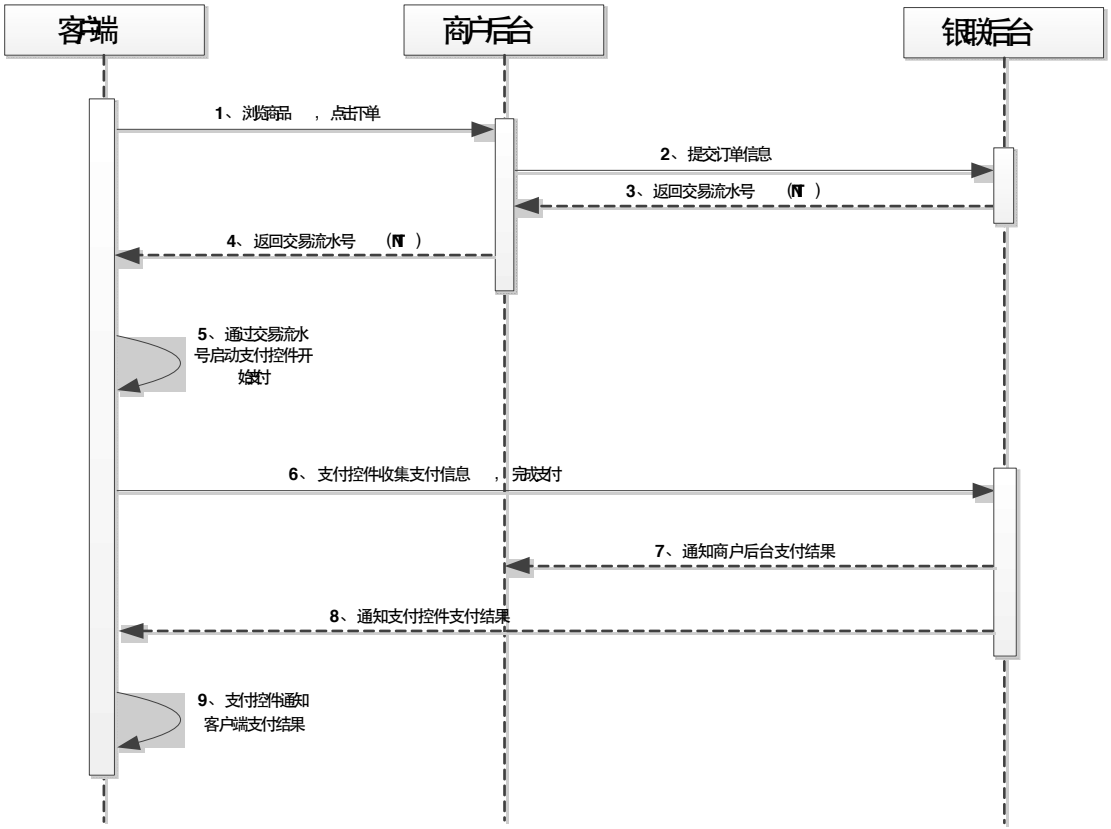
1. 概述

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。

(温馨提示：对于使用中国银联老版支付控件商户可以跳过其他章节，直接对照本文档第5部分对工程进行改造)

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（TN），将交易流水号返回至客户端；

（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）银联后台同时也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至客户端；

注: 本文档主要关注上述流程中（5）（9）部分的实现

iOS版本支付控件适用iOS 6.0及以上版本终端设备。

3. 测试帐号

以下是测试用卡号、手机号等信息（此类信息仅供测试使用，不会发生真实交易）

招商银行借记卡：6226090000000048 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. iOS客户端

本小节涉及到SDK的接口说明、Xcode工程配置及其接口调用细节，需要读者具有

一定iOS编程经验。

4.1. SDK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_iphone/paymentcontrol，以下部分提及的文件均在该目录中。银联支付控件静态库，以下简称UPPaymentControl，包含以下两个文件：

UPPaymentControl.h

libPaymentControl.a

4.2. 接口说明

1. 支付接口

```
- (BOOL)startPay:(NSString*)tn
      fromScheme:(NSString*)schemeStr
            mode:(NSString*)mode
  viewController:(UIViewController*)viewController
```

各个参数定义如下表：

参数名称	类型	含义
tn	NSString*	必填项； 交易流水号，商户后台向银联后台提交订单信息后，由银联后台生成并下发给商户后台的交易凭证；
schemeStr	NSString *	必填项; 商户自定义协议，商户在调用支付接口完成支付后，用于引导支付控件返回而定义的协议，具体请参考4.4第二步中URL Type定义;

mode	NSString*	必填项； 接入模式，标识商户以何种方式调用支付控件，该参数提供以下两个可选值： "00"代表接入生产环境（正式版本需要）； "01"代表接入开发测试环境（测试版本需要）；
viewController	UIViewController*	必填项； 发起调用的视图控制器，商户应用程序调用银联手机支付控件的视图控制器；
返回值	BOOL	YES：调起支付控件成功； NO：调起支付控件失败；

2、检查是否安装银联App的接口

```
- (BOOL)isPaymentAppInstalled
```

此函数无传入参数，主要功能检测用户手机上是否安装银联支付的APP，当用户手机上安装了银联支付APP时候返回YES。

参数名称	类型	含义
返回值	BOOL	YES：已安装银联支付APP； NO：未安装银联支付APP；

3、返回结果接口

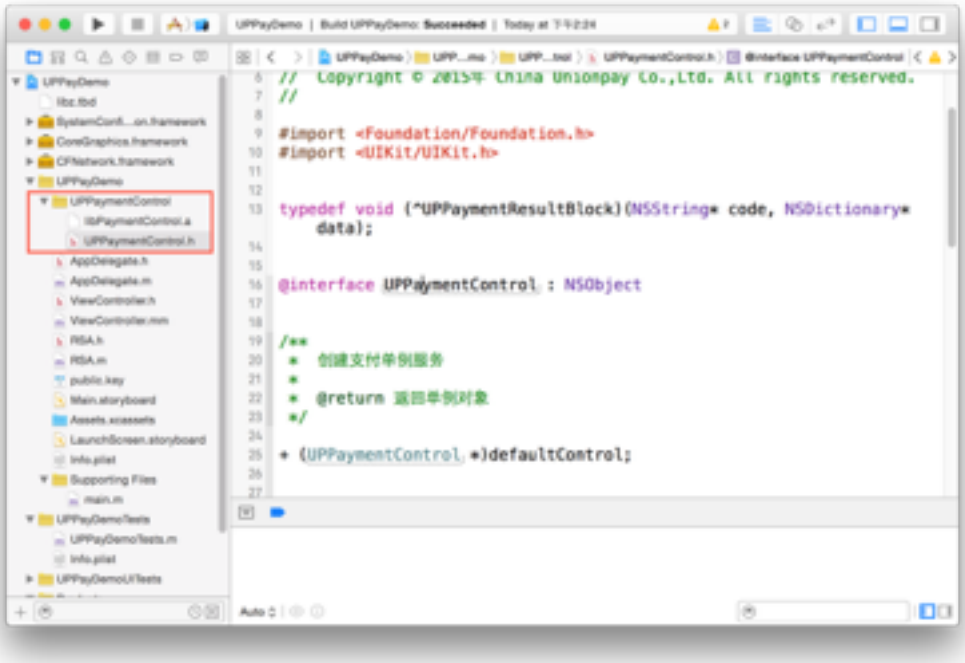
```
- (void)handlePaymentResult:(NSURL*)url
                completeBlock:(UPPaymentResultBlock)completionBlock;
```

各参数定义如下表：

参数名称	类型	含义
url	NSURL*	必填项； 支付结果url，传入后由SDK解析并通过completionBlock回调商户客户端；
completionBlock	Block	必填项； 商户客户端定义的结果处理方法，此方法包行code、data两个传入参数，本文档4.5第三部分将详细说明；

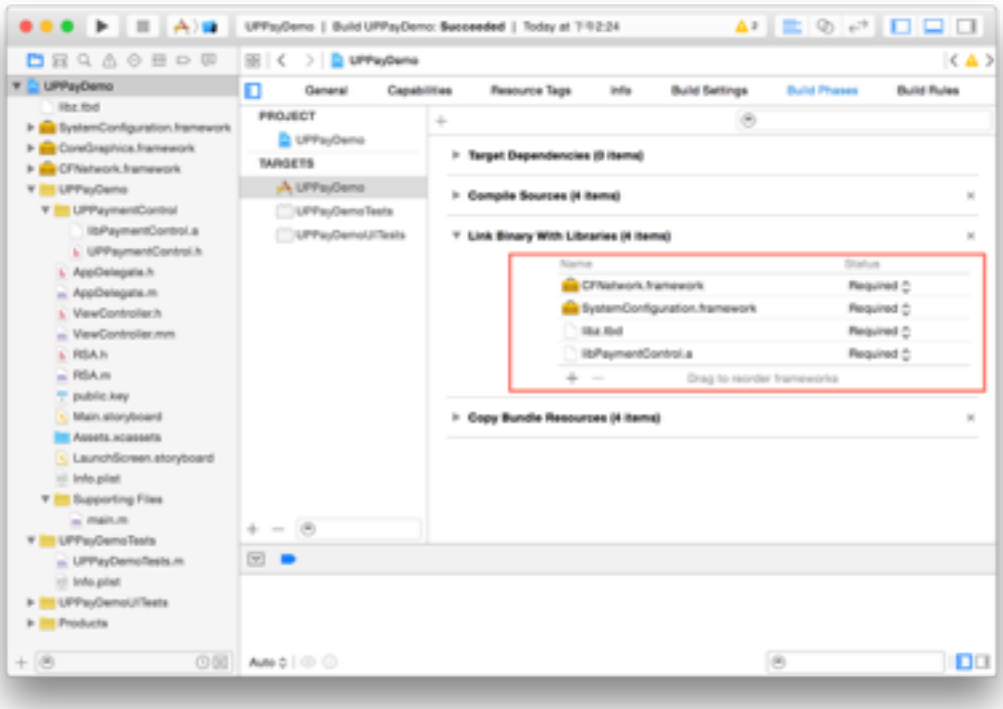
4.3. 添加SDK包

使用UPPaymentControl需要将paymentcontrol/inc目录下UPPaymentControl.h文件和paymentcontrol/libs目录下的libPaymentControl.a文件添加到商户应用的工程中，添加后如图：

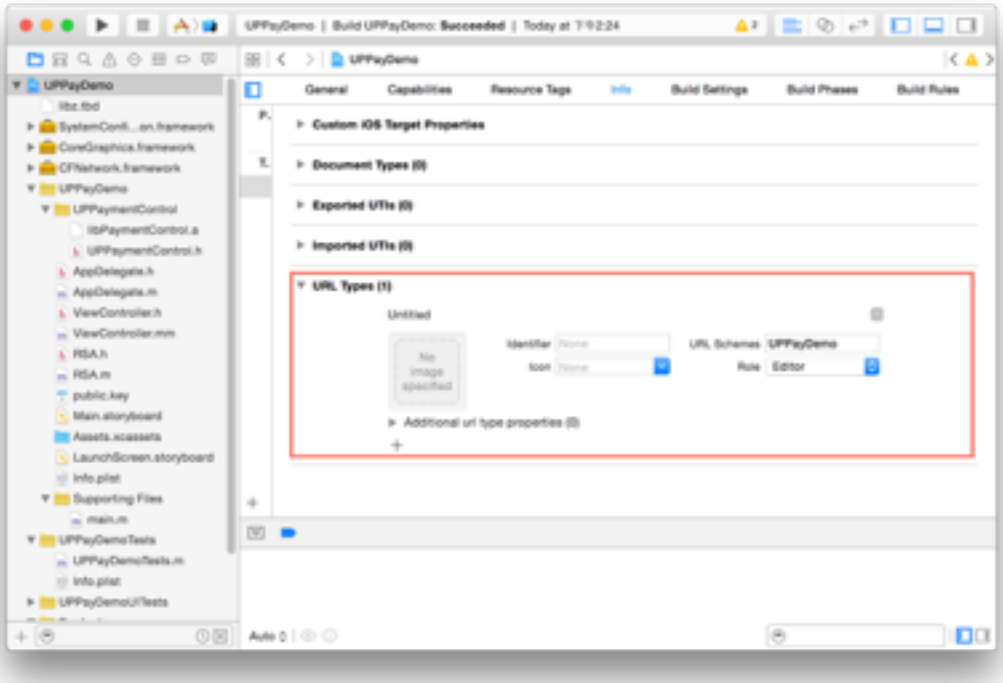


4.4. 工程配置

1. 使用UPPaymentControl需要添加CFNetwork.framework、SystemConfiguration.framework 、libz、libPaymentControl.a到工程中，添加后如下图：



2、在工程info.plist设置中添加一个URL Types回调协议(在UPPayDemo工程中使用“UPPayDemo”作为协议)，用于在支付完成后返回商户客户端。请注意URL Schemes需要是唯一的。



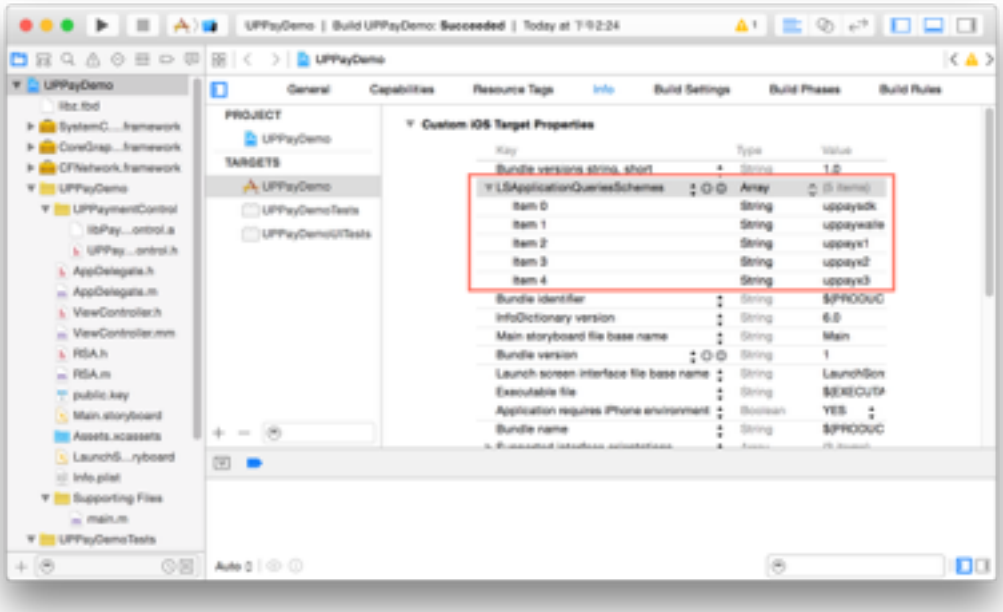
3、http请求设置（ats）

在测试环境测试时，需要在工程对应的plist文件中添加NSAppTransportSecurity Dictionary 并同时设置里面NSAllowsArbitraryLoads 属性值为 YES，具体设置可参照以下截图：

发生产环境可删除此设置。向Apple发布正式版本时请删除此设置。

4、添加协议白名单

在Xcode7.0之后的版本中进行开发，需要在工程对应的plist文件中，添加LSApplicationQueriesSchemes Array并加入uppaysdk、uppaywallet、uppayx1、uppayx2、uppayx3五个item，具体设置可参考以下截图：



或者直接添加如下代码到plist文件中：

```
<key>LSApplicationQueriesSchemes</key>
     <array>
         <string>uppaysdk</string>
         <string>uppaywallet</string>
         <string>uppayx1</string>
         <string>uppayx2</string>
         <string>uppayx3</string>
     </array>
```

4.5. 调用插件

在需要调用支付控件接口的代码文件内引用头文件UPPaymentControl.h。

（注意：如果工程的compile source as 选项的值不是Objective–C++，则引用此头文件的文件类型都要改为.mm）

1、支付接口调用

商户App从商户服务器获取tn，当tn不为空时，调用支付接口。

```
//当获得的tn不为空时，调用支付接口
 if (tn != nil && tn.length > 0)
 {
        [[UPPaymentControl defaultControl]
                startPay:tn
              fromScheme:@"UPPayDemo"
                    mode:self.tnMode
      viewController:self];
 }
```

2、检测是否已安装银联App接口调用

```
if([[UPPaymentControl defaultControl] isPaymentAppInstalled])
{
    //当判断用户手机上已安装银联App，商户客户端可以做相应个性化处理
}
```

3、返回结果接口调用

支付控件结果处理函数handlePaymentResult: completeBlock:需要在工程AppDelegate文件的application: openURL: options: 方法中进行调用。

支付控件结果处理函数handlePaymentResult: completeBlock:包含两个参数，参数1url为支付结果串，由handlePaymentResult: completeBlock:方法解析url内容；参数2completionBlock为商户APP定义的结果处理方法，包含两个传入参数code和data，其中code表示支付结果，取值为suceess、fail、cancel分别表示支付成功、支付失败、支付取消，data表示结果签名数据，商户可使用银联公钥验证结果真实性。

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，建议无视结果中的内容和签名信息，直接到自己的后台查状态，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；如果需要在APP中验签，则需要自行实现验签公钥更新的机制，否则银联更新密钥后会验签失败。

调用支付接口后，结果处理方法示例代码：

```
    - (BOOL)application:(UIApplication *)app openURL:(NSURL *)url options:
(NSDictionary<NSString*, id> *)options {
        [[UPPaymentControl defaultControl] handlePaymentResult:url
completeBlock:^(NSString *code, NSDictionary *data) {

        if([code isEqualToString:@"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
            if(data != nil){
                //数据从NSDictionary转换为NSString
                NSData *signData = [NSJSONSerialization dataWithJSONObject:data
                                                                   options:0
                                                                     error:nil];
                NSString *sign = [[NSString alloc] initWithData:signData
encoding:NSUTF8StringEncoding];

                //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([self verify:sign]) {
                    //验签成功
                }
                else {
                    //验签失败
                }
            }

            //结果code为成功时，去商户后台查询一下确保交易是成功的再展示成功
        }
        else if([code isEqualToString:@"fail"]) {
            //交易失败
        }
        else if([code isEqualToString:@"cancel"]) {
            //交易取消
        }
    }];

    return YES;
  }
```

completeBlock中的NSDictionary *data结构如下：

```
sign —— 签名后做Base64的数据
data —— 用于签名的原始数据，结构如下：
         pay_result —— 支付结果success，fail，cancel
         tn          —— 订单号
```

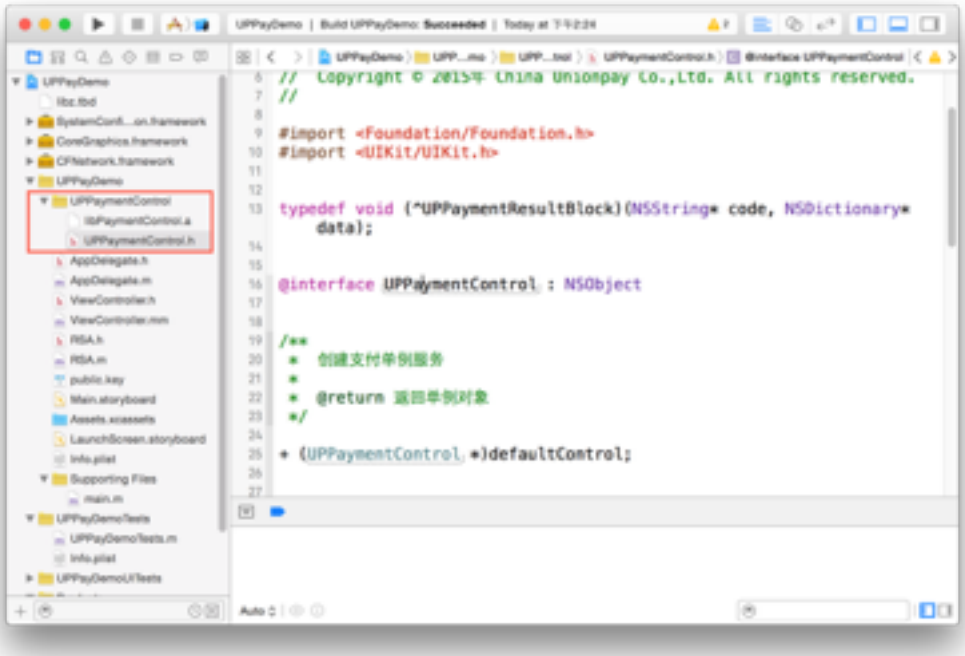
Data转换为String后的示例如下：

```
{  "sign" : "Xo/pgkzSJSlRTX2e+CjW/k1IjIV1newqfb7p1sDIpK/
yPQv9p1jQAdAdKwhBwtyjO3tkFC6I2aLcTaxLHlYQx6/xw9QE0eumkVqAhypk/
VyoDWZXxWske+EcduwEkBTxyIgA0ZsbKlpS1JxsciOc6bT+f36jTLa05ZAKZTVErg9sAG3wMjae1TyKd2511
Rvvi+tuihYgOmwuMnKzrqksEyqc69wloqi34qx0YqFolMeqQ1UfoglUhZy6s2s4ChKcxHjAFjp/rU/
7iHudjAIGtO7+ySahArmw6ltuIxFWYEvpn5xI3Ceur1d11NBphK62it7kBZ1laxUFI98DzalVFQ==",
"data" : "pay_result=success&tn=89939408566062273670 1&cert_id=68759585097"}
```

5. 老版控件用户修改说明

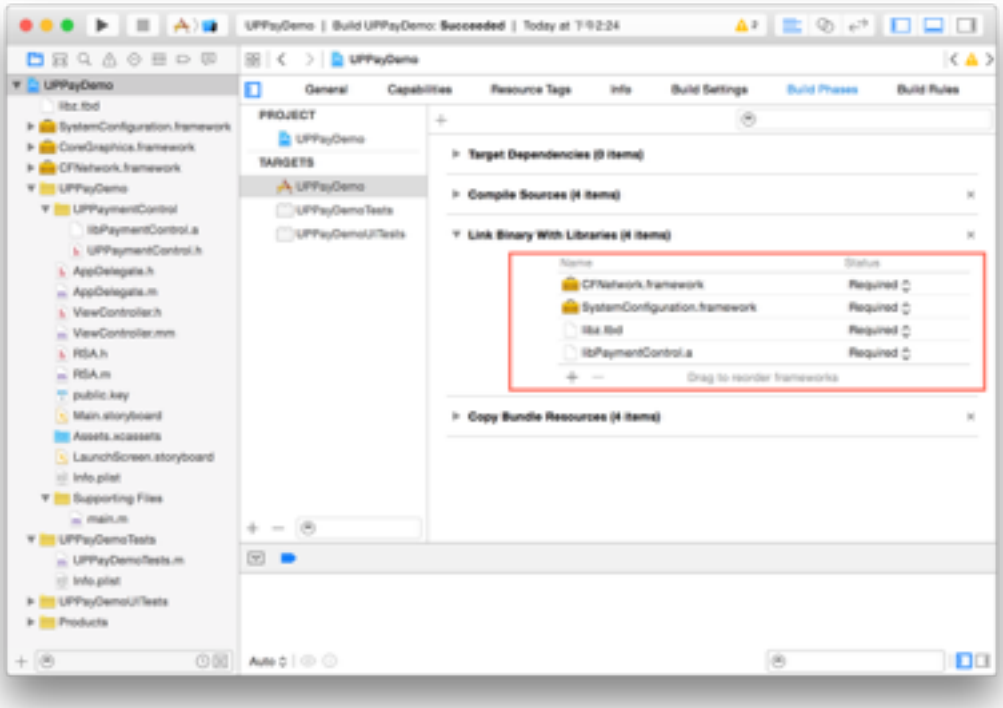
5.1. 添加SDK包

使用老版本银联支付控件的商户请先将老版本银联支付控件从工程中删除，再将新paymentcontrol/inc目录下UPPaymentControl.h文件和paymentcontrol/libs目录下libPaymentControl.a文件添加到商户应用的工程中；添加后如图：

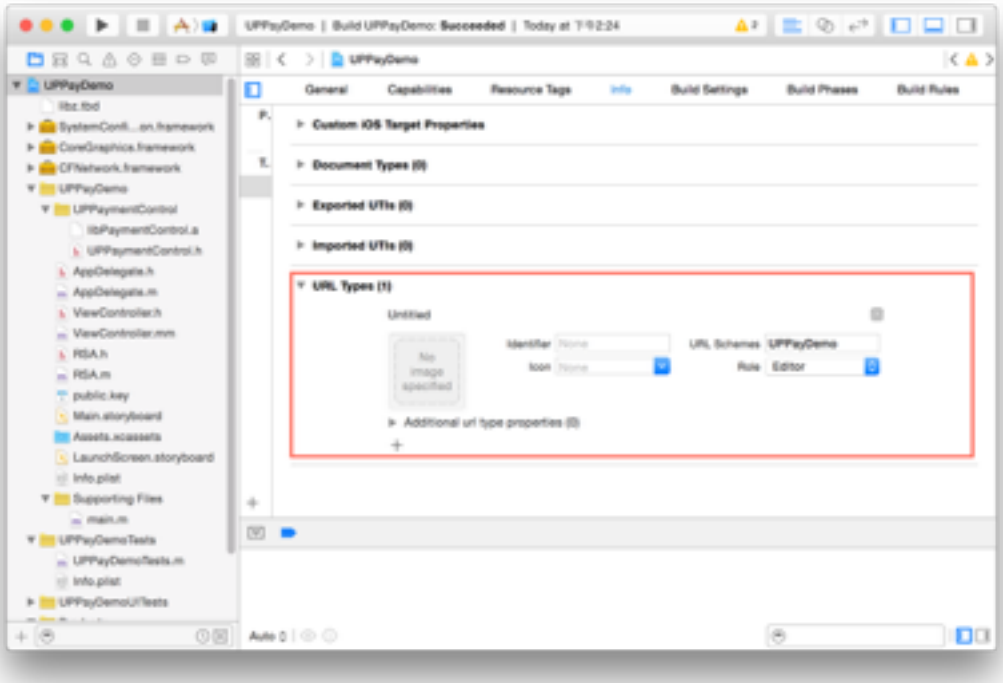


5.2. 工程配置

1、使用老版本银联支付控件的商户请先将工程配置中老版支付控件对应的三方库删除，再添加CFNetwork.framework、SystemConfiguration.framework 、libPaymentControl.a、libz到工程中；添加后如下图：



2、在工程info.plist设置中添加一个URL Types回调协议(在UPPayDemo工程中使用“UPPayDemo”作为协议)，用于在支付控件完成支付后返回到商户客户端。请注意URL Schemes需要是唯一的。



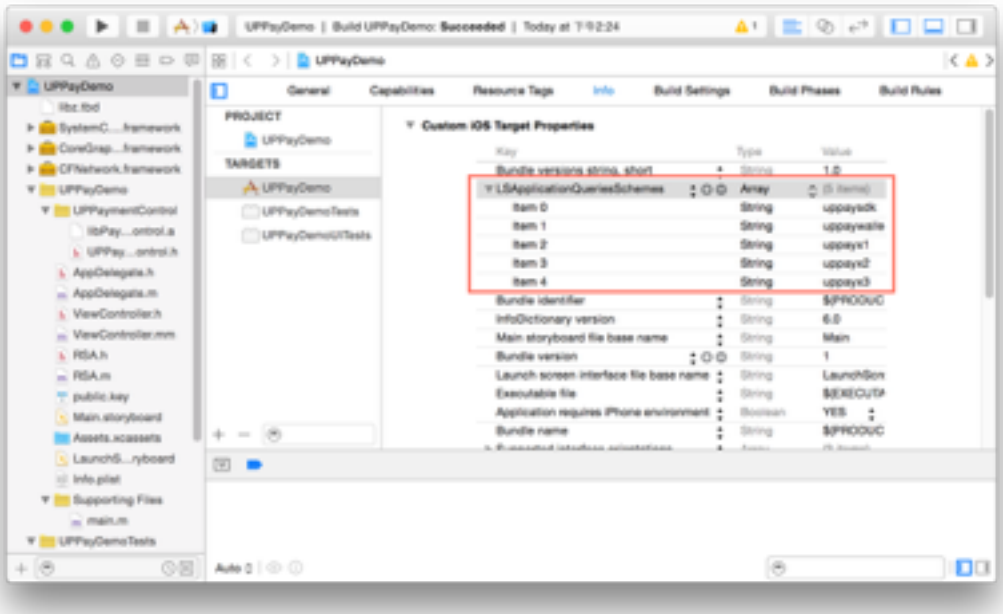
3、http请求设置（ats）

在测试环境测试时，需要在工程对应的plist文件中添加NSAppTransportSecurity Dictionary 并同时设置里面NSAllowsArbitraryLoads 属性值为 YES，具体设置可参照以下截图：

发生产环境可删除此设置。向Apple发布正式版本时请删除此设置。

4、添加协议白名单

在工程对应的plist文件中，添加LSApplicationQueriesSchemes Array并加入uppaysdk、uppaywallet、uppayx1、uppayx2、uppayx3五个item，具体设置可参考以下截图:



或者直接添加以下代码到plist文件中：

```
<key>LSApplicationQueriesSchemes</key>
     <array>
         <string>uppaysdk</string>
         <string>uppaywallet</string>
         <string>uppayx1</string>
         <string>uppayx2</string>
         <string>uppayx3</string>
     </array>
```

5.3. 接口说明和调用插件

请参考4.2接口说明和4.4调用插件说明。

6. 常见问题

请参见https://open.unionpay.com帮助中心-FAQ