

# 手机支付控件指南

Apple Pay 版

V1.0.0

中国银联

## 版本信息

版本号	日期	说明
0.4.0	2015-12-19	初稿
1.0.0	2016-01-08	1、增加随机优惠描述及其界面示例。 2、增加新的结果返回值，UPPaymentResultStatusUnknownCancel，表示用户发起支付请求后，再点击取消时对应的支付结果状态。

## 目录

1	概述 .....	1
2	支付流程介绍.....	1
2.1	展示 Apple Pay 按键的判断 .....	1
2.2	支付控件调用 .....	2
3	iOS 客户端接入说明.....	3
3.1	SDK 说明.....	3
3.2	接口说明.....	3
3.3	添加 SDK 包 .....	6
3.4	工程配置.....	6
3.5	调用插件.....	7
4	常见问题总结.....	10

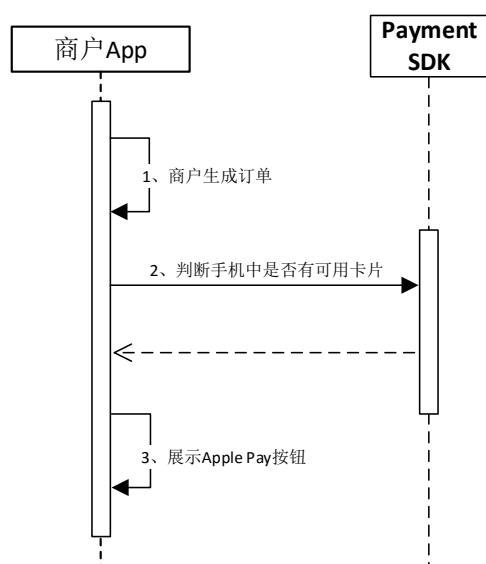
## 1 概述

银联手机支付控件 Apple Pay 版，主要为合作商户提供基于 iPhone 手机客户端的 Apple Pay 支付服务。

## 2 支付流程介绍

### 2.1 展示 Apple Pay 按键的判断

商户 App 在调用 Apple Pay 之前要注意根据苹果 PaymentSDK 提供的系统函数判断手机是否支持 Apple Pay，从而判断是否显示 Apple Pay 支付按钮，流程如下：



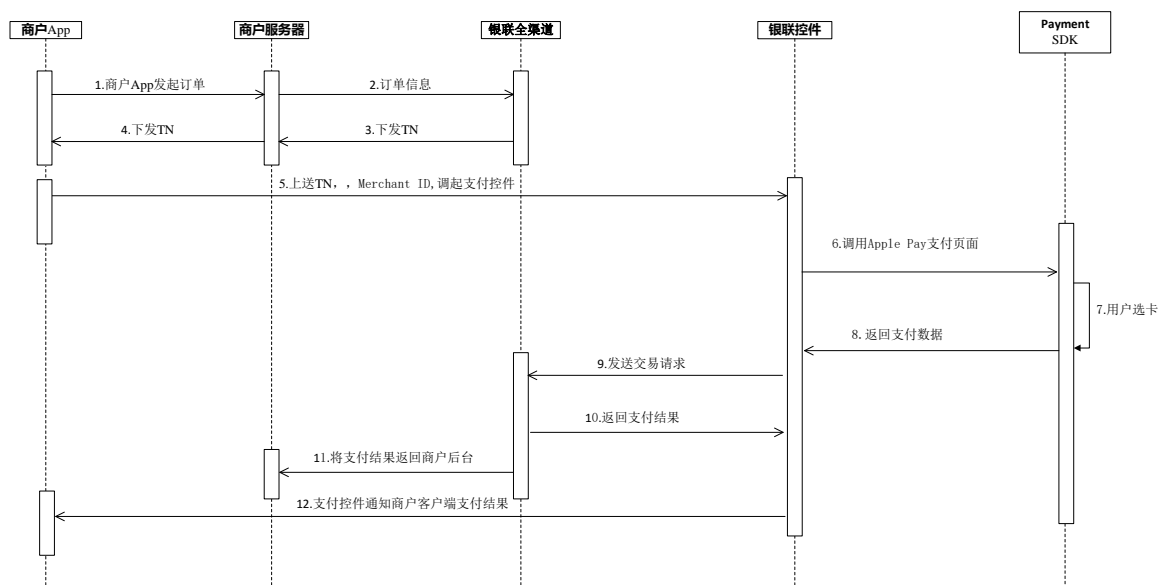
1、用户购买商品或服务，商户生成订单；

2、判断手机是否支持 Apple Pay 功能，以及是否已加载有可用的支付卡片；

判断方法为：

```
if([PKPaymentAuthorizationViewController  
canMakePaymentsUsingNetworks:@[PKPaymentNetworkChinaUnionPay]])  
{  
    //展示苹果支付图标  
}
```

## 2.2 支付控件调用



流程图说明：

- (1) 用户在客户端中点击 Apple Pay 按钮，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，同时传入 MerchantID；
- (6) 支付控件调起支付页面；
- (7) 用户在支付页面中选择银行卡，验证用户信息；
- (8) 返回验证后的支付数据；

- ( 9 ) 银联支付控件上送交易；
  - ( 10 ) 返回交易结果至银联支付控件；
  - ( 11 ) 银联全渠道平台将成功交易结果返回商户后台；
  - ( 12 ) 银联支付控件把支付结果返回给商户客户端，商户客户端向用户展示支付结果。
- 若交易失败，应根据支付控件返回的 `errorDescription` 错误信息描述，在商户客户端内提示给用户。若返回的结果中包含 `otherInfo` 字段，表明此笔交易参与了立减折扣，商户应将折扣信息在支付成功界面展示给用户。

## 3 iOS 客户端接入说明

本小节提供给那些具有一定 iOS 编程经验和了解面向对象概念的读者使用。

SDK 文件所在目录：upmp\_iphone/sdk，以下部分所说文件，均在该目录中。

### 3.1 SDK 说明

Apple Pay 版本静态库，以下简称 UPAPayPlugin，包含文件：

UPAPayPlugin.h

UPAPayPluginDelegate.h

libUPAPayPlugin.a

### 3.2 接口说明

## 1、支付接口

```

+ (BOOL)startPay:(NSString*)tn

                mode:(NSString*)mode

viewController:(UIViewController*)viewController

                delegate:(id<UPAPayPluginDelegate>)delegate

andAPMerchantID:(NSString*)mID

```

各个参数定义如下表：

参数名称	类型	含义
tn	NSString*	<b>必填项；</b> 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；
mode	NSString*	<b>必填项；</b> 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*	<b>必填项；</b> 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；
delegate	id<UPAPayPluginDelegate>	<b>必填项；</b> 实现 UPAPayPluginDelegate 方法的 UIViewController；
mID	NSString*	<b>必填项；</b> 苹果公司分配的商户号，表示调用 Apple Pay 所需要的 MerchantID；
返回值	BOOL	YES：调起支付控件成功； NO：调起支付控件失败；

## 2、结果回调接口

```
-(void) UPAPayPluginResult:(UPPayResult *) payResult;
```

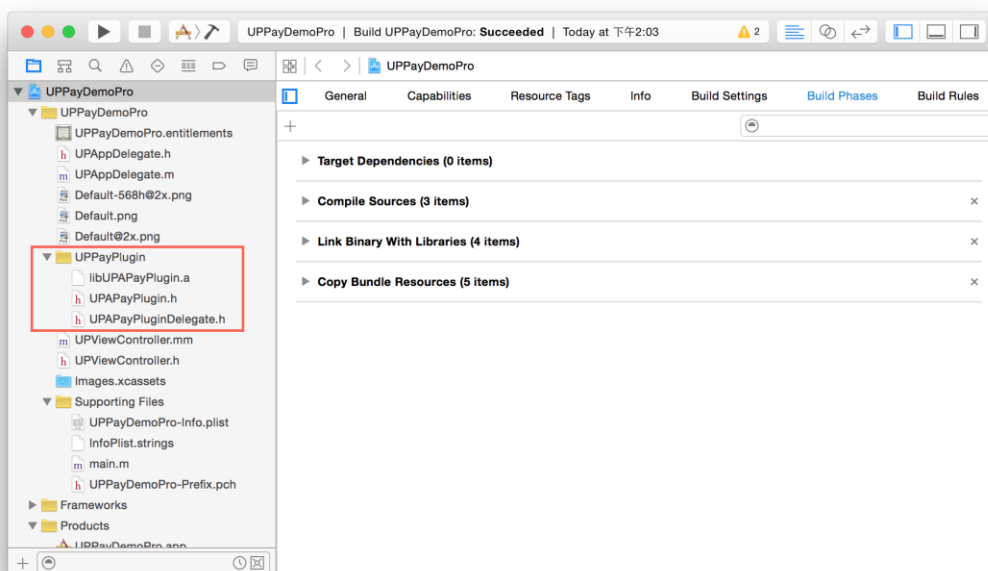
各个参数定义如下表：

参数名称	类型	含义
payResult	UPPayResult*	<b>回调时必返参数；</b> 支付结果信息，UPPayResult 中包含支付成功，支付失败，支付取消三种状态及其错误信息和优惠信息，详情参考 3.5 第 2 部分处理支付结果；



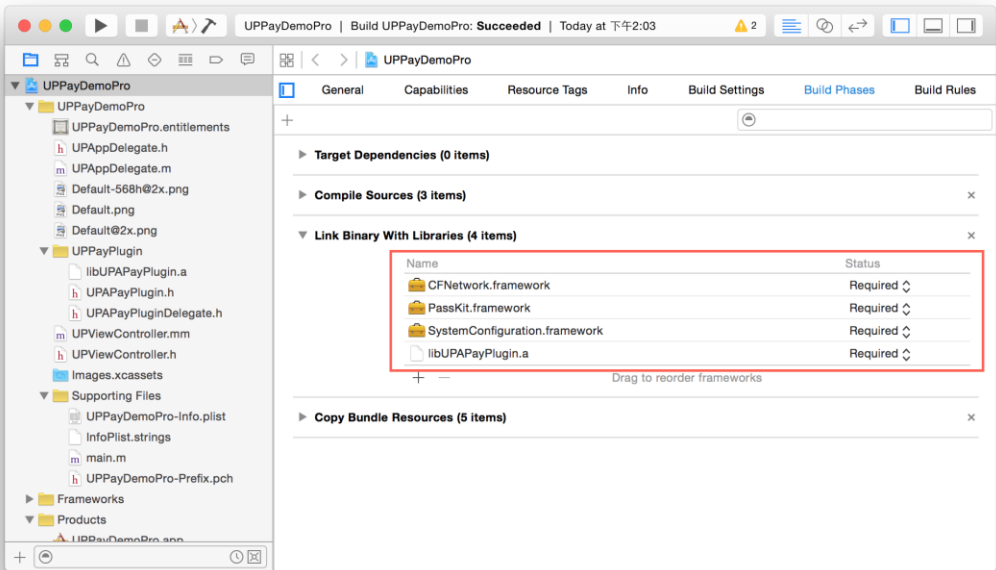
### 3.3 添加 SDK 包

将 sdk/inc 目录和 sdk/libs 目录下对应版本的三个文件添加到商户 App 工程中，添加后如图：



### 3.4 工程配置

- 1、添加 CFNetwork.framework、libUPAPayPlugin.a、PassKit.framework、SystemConfiguration.framework 到商户 App 工程中，添加后如下图：



2、http 请求设置

在 Xcode7.0 之后的版本中进行 http 请求时，需要在工程对应的 plist 文件中添加 NSAppTransportSecurity Dictionary 并同时设置 NSAllowsArbitraryLoads 属性值为 YES，具体设置可参照以下截图：

UPPayDemoPro-Info.plist	NSAppTransportSecurity	Dictionary	(1 item)
InfoPlist.strings	NSAllowsArbitraryLoads	Boolean	YES

3.5 调用插件

在需要调用支付控件的源文件内引入头文件 UPAPayPlugin.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)
{
    if([PKPaymentAuthorizationViewController
canMakePaymentsUsingNetworks:@[PKPaymentNetworkChinaUnionPay]])
    {
        //调用银联支付控件
        [UPAPayPlugin startPay:tn mode:self.testMode viewController:self delegate:self
andAPMerchantID:kAppleMerchantID];
    }
}

```

## 2、处理支付结果

银联手机支付控件结果回调函数有四个支付状态返回值，在 UPPayResult 类中有关于 UPPaymentResultStatus 的详细说明。商户 App 通过在工程中添加头文件“UPAPayPluginDelegate.h”，然后在处理交易结果界面中根据该头文件中定义的回调函数：-(void)UPAPayPluginResult:(UPPayResult \*)result 来实现回调方法，从而根据支付结果的不同进行相应的处理。

UPPayResult 类结构：

```

@interface UPPayResult:NSObject
@property UPPaymentResultStatus paymentResultStatus;
@property (nonatomic,strong) NSString* errorDescription;
@property (nonatomic,strong) NSString* otherInfo;
@end

```

1) UPPaymentResultStatus，表示四个支付状态返回值，结构如下：

```

typedef NS_ENUM(NSInteger,UPPaymentResultStatus) {

    UPPaymentResultStatusSuccess,        //支付成功
    UPPaymentResultStatusFailure,        //支付失败
    UPPaymentResultStatusCancel,         //支付取消
    UPPaymentResultStatusUnknownCancel  //支付取消，交易已发起，状态不确定，商户需查询商户后台确认支付状态
}

```

```
};
```

2) `errorDescription`，表示支付失败时候服务器返回的错误描述，包括文字信息与应答码两部分。

**例如：**`errorDescription` 字段内容为“可用余额不足 [ 1000051 ]”，此信息前半部分为文字错误信息，后 7 位为错误应答码。当支付成功或支付取消的时候 `errorDescription` 取值为 `nil`。

3) `otherInfo`，目前表示成功支付时包含的优惠信息。

**例如：**`otherInfo` 为“`currency=元&order_amt=20.00&pay_amt=15.00`”，其中 `currency` 表示币种，`order_amt` 表示订单金额，`pay_amt` 表示实付金额。

优惠形式包括但不限于立减折扣，随机折扣等。当无优惠信息时 `otherInfo` 取值为 `nil`，可不作处理。当出现优惠信息时，商户可以根据此优惠信息组织个性化的优惠页面进行展示，但应在页面中加入**原金额、优惠金额及“云闪付”**要素，UI 示例如下：



示例 1



示例 2

商户订单是否成功支付应以商户后台收到全渠道返回的支付结果为准，此处支付控件返回结果仅作为商户 App 向用户展示支付结果使用。

结果回调函数示例代码：

```
if(result.paymentResultStatus == UPPaymentResultStatusSuccess) {
    NSString *otherInfo = result.otherInfo?result.otherInfo:@"";
    NSString *successInfo = [NSString stringWithFormat:@"支付成功\n%@",otherInfo];
    [self showAlertMessage:successInfo];
}
else if(result.paymentResultStatus == UPPaymentResultStatusCancel){

    [self showAlertMessage:@"支付取消"];
}
else if (result.paymentResultStatus == UPPaymentResultStatusFailure) {

    NSString *errorInfo = [NSString stringWithFormat:@"%s",result.errorDescription];
    [self showAlertMessage:errorInfo];
}
else if (result.paymentResultStatus == UPPaymentResultStatusUnknownCancel) {

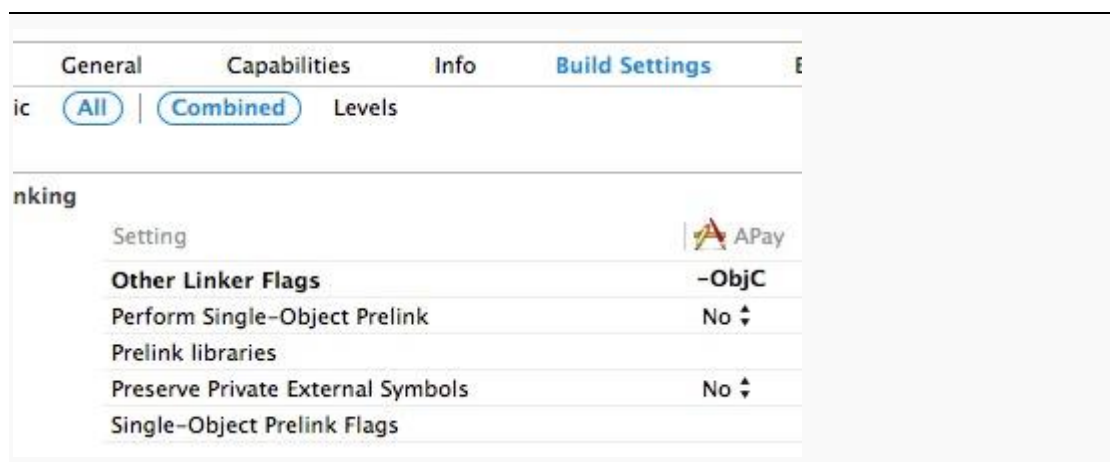
    //TODO UPPaymentResultStatusUnknownCancel表示发起支付以后用户取消，导致支付状态不确认，需要查询商户后台确认真实的支付结果
    NSString *errorInfo = [NSString stringWithFormat:@"支付过程中用户取消了，请查询后台确认订单"];
    [self showAlertMessage:errorInfo];
}
```

## 4 常见问题总结

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

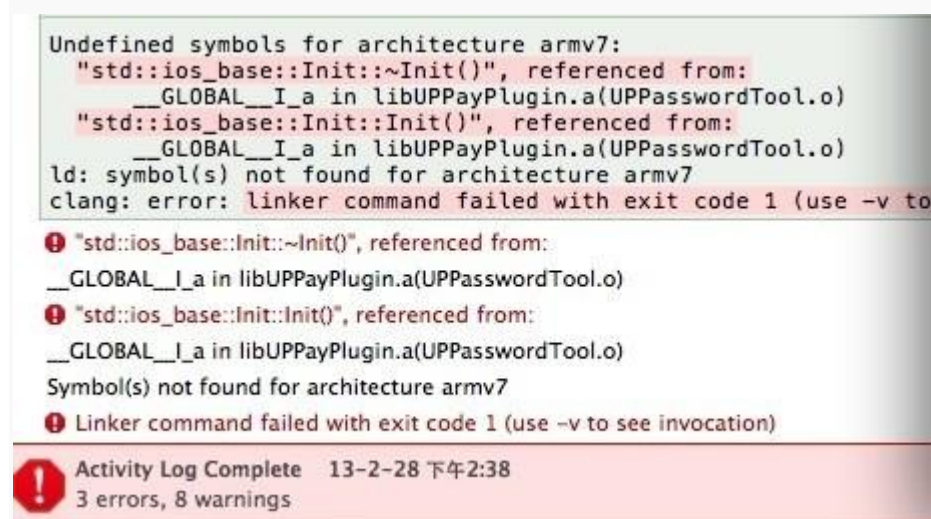
### 4.1 添加-ObjC 宏

选择工程 targets->build settings ->Linking->other linker flags



## 4.2 编译时提示 Undefined for architecture XXX 错误

如：



1) 由于支付控件使用到了 C、C++ 和 OC 混编的情况，所以商户工程引入 UPAPayPlugin.h 头文件以后可能会出现链接错误。这个时候可以通过两种方式解决：

- ① 将涉及到引用 UPAPayPlugin.h 的源文件的后缀名都改为.mm；
- ② 如果商户不想修改源文件的后缀名，可以在工程中添加一个空的继承自 NSObject 的类，并将文件.m 后缀名该改为.mm 即可。方法为 new file->Objective-C class->类名自取->保存->修改后缀名为.mm；

③ 将工程的 compile source as 选项的值设置为 Objective-C++ ；

2 ) 由于在 UPPayDemo 工程中添加了自定义的库文件 libUPAPayPlugin.a , 当编译 Demo 工程时 , 应该检查工程设置 Search Paths 里的 Framework Search Paths、Header Search Paths、Library Search Paths 的路径设置 , 看设置路径是否正确 , 另外还要注意里边是否多余一些不确定的路径 ;

3 ) 将 xcode 工程中 Build Setting -> Apple LLVM 7.0-Language-C++ 标签下的 C++ Language Dialect 和 C++ Standard Library 属性值修改为 Compiler Default。如仍无法解决 , 尝试把 other linker flags 选项的值设为 -force\_load+空格+控件路径 , 如 : -force\_load \$(PROJECT\_DIR)/libUPAPayPlugin.a。如果报错 , 上下文应该会有 libUPAPayPlugin.a 文件找不到的异常信息 , 比如

ld: file not found: /Users/apple/Desktop/Communication 2/libUPAPayPlugin.a

clang: error: linker command failed with exit code 1 (use -v to see invocation)

请确定 libUPAPayPlugin.a 文件确实存在于此路径 , 是否因为上下文路径配置错误。

### 4.3 控件闪退异常'NSInvalidArgumentException', reason: '-[\_\_NSCFConstantString newSizeWithFont:详略]

'NSInvalidArgumentException', reason: '-[\_\_NSCFConstantString newSizeWithFont:详略]:

unrecognized selector sent to instance 0x[随机数]

如图 :

```

2015-03-20 09:41:25.817 [213:60b] -[__NSCFConstantString newSizeWithFont:forWidth:lineBreakMode:]:
unrecognized selector sent to instance 0x708b24
2015-03-20 09:41:25.818 [213:60b] *** Terminating app due to uncaught exception
'NSInvalidArgumentException', reason: '-[__NSCFConstantString newSizeWithFont:forWidth:lineBreakMode:]:
unrecognized selector sent to instance 0x708b24'
*** First throw call stack:
(0x2d9c5f0b 0x38158ce7 0x2d9c9837 0x2d9c8137 0x2d9c93c8 0x2cd7cd 0x2cdbbb 0x2b8d35 0x2b86b1 0x2e4a61
0x2cb937 0x2c8eb3 0x301f9a53 0x302a430d 0x302a4223 0x302a3801 0x302a3529 0x302a3299 0x302a3231
0x301f5305 0x2fe7131b 0x2fe6cb3f 0x2fe6c9d1 0x2fe6c3e5 0x2fe6c1f7 0x2fe65f1d 0x2d991039 0x2d98e9c7
0x2d98ed13 0x2d8f9769 0x2d8f954b 0x328566d3 0x30258891 0x19f5c1 0x38656ab7)
libc++abi.dylib: terminating with uncaught exception of type NSException
(lldb) po 0x708b24

```

如果把 other linker flags 选项的值设为-ObjC 仍不能解决，可尝试替换为

-force\_load+空格+控件路径，如：-force\_load \$(PROJECT\_DIR)/libUPAPayPlugin.a

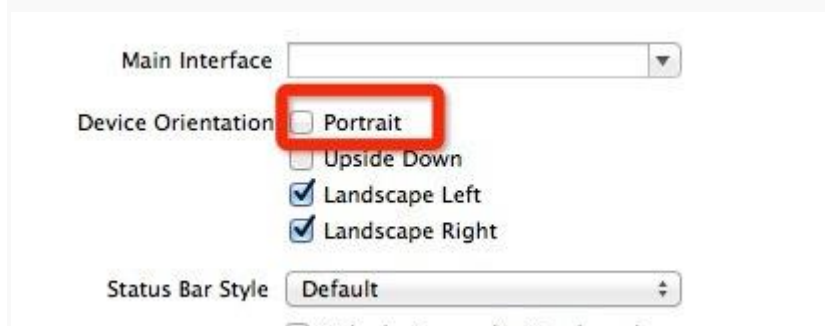
## 4.4 orientation 异常

详细现象为控件崩溃，报错信息：Terminating app due to uncaught exception

'UIApplicationInvalidInterfaceOrientation'，reason: 'Supported orientations has no common orientation with the application, and shouldAutorotate is returning YES'

请勿修改横竖屏配置，iphone 控件只能竖屏，ipad 控件才支持横竖屏。

如果 app 本身为横屏，请代码控制横竖屏，勿修改配置。



## 4.5 其他第三方静态库（如支付宝、微信等）的文件产生冲突 duplicate symbols for architecture XXX

如：



```
ld: 116 duplicate symbols for architecture armv7
clang: error: linker command failed with exit code 1 (use -v to see invocation)
```

```
116 duplicate symbols for architecture armv7
```

❗ Linker command failed with exit code 1 (use -v to see invocation)



Activity Log Complete 14-6-4 16:14

1 error, 49 warnings

控件指南中提到的添加-ObjC 宏的地方修改为 `-force_load+空格+控件路径`，如：

`-force_load $(PROJECT_DIR)/libUPAPayPlugin.a`，不会影响支付宝插件接入。

## 4.6 进入控件时界面无限加载

1. 调用控件的逻辑放在主线程中执行。因为在子线程中不允许操作 UI，无限加载原因是子线程死锁。
2. 参考本指南“添加 SDK 包”中“添加-ObjC 宏”部分，注意大小写拼写错误。（此步骤操作有问题时也可修改为 `-force_load+空格+控件路径`，如：`-force_load $(PROJECT_DIR)/libUPAPayPlugin.a`）
3. Build Setting -> Apple LLVM compiler Language-C++：



## 4.7 Attempt to present XX on XX while a presentation is in progress!

Warning: Attempt to present <UPNavController: 0x8bc24d0> on <tttViewC  
ontroller: 0x8b9a390> while a presentation is in progress!

animation 完成的时候再调用支付控件，两个 present 同时调用会出现警告。