

同步移动开发平台SDK接入 参考手册

V3.1

版本	更新说明	时间
1.0	1、支持同步账号登录，注册，找回密码。 2、支持支付宝，财付通支付方式。	
2.0	1、新增直充方式。 2、新增神州行支付方式。 3、优化速度。	
2.1	1、支持iPhone5、iOS6。 2、支持横竖屏翻转。	2012-10-23
2.2	优化统计	2012-11-7
2.3	1、新增游戏中心，带有完整的个人中心，充值中心，游戏推荐频道，游戏论坛等功能。 2、新增登录成功session验证。 3、修改部分接口。	2013-4-19
2.4	1、添加新一套接口，兼容2.3旧接口。 2、添加平台界面方向设定、自动旋转开关接口。 3、移除用户中心中的充值模块，进入用户中心接口改动。（查看接口改动说明） 4、添加armv7s架构支持。 5、添加玩家登录的情况下进行直充的充值记录。 6、静态库由.a文件改为Framework。 7、资源文件改为bundle形式。 8、添加调试模式、更新检查。	2013-6-4
3.0	1、不再兼容2.4之前的旧接口。 2、重新设计UI界面。 3、加入平台中间货币（推币）。 4、添加银联、mo9等充值渠道。 5、提供订单结果查询接口及物品购买结果回调。 6、添加一键注册功能，注册可使用非邮箱账号。 7、添加微博登录功能； 8、添加单设备上的自动登录功能； 9、充值调试改为使用测试账号，调试模式可调试检测更新。	2013-8-12

3.0.1	<p>1、新增平台初始化方法，该方法自动完成检查更新工作及回调处理，强烈建议使用；</p> <p>2、接入SDK调试的各个环节添加NSLog信息打印；</p> <p>3、获取推币余额、支付改为异步，添加等待提示，购买失败回调错误类型细分，添加网络不流畅的错误分类；</p> <p>4、更新银联支付SDK (需要添加Security.framework)；</p> <p>5、添加微博登录邮箱后缀提示功能；</p> <p>6、微博登录成功后保存最后一次登录账号；</p> <p>7、短代支付快速生成原生短信支持（需要添加MessageUI.framework)；</p> <p>8、添加离开平台类型参数（于userInfo字典中）。</p>	2013-9-26
3.1	<p>1、全面适配iOS7，添加iOS7特性开关（使用iOS7以上SDK编译工程时打开SDK中的开关：TBSetSupportiOS7:)；</p> <p>2、更换银联SDK，适配iOS7 (需添加AdSupport.framework)；</p> <p>3、添加公告消息模块；</p> <p>4、添加支付宝快捷支付、招商银行两个充值渠道；</p> <p>5、界面细节及SDK稳定性修复。</p>	2013-12-25

目录

开发前需知	5
一、SDK构成	5
二、开发环境搭建	6
1、工程配置	6
2、添加Framework.....	6
1. 添加系统Framework和链接库	6
2. 添加同步SDK的Framework	7
3、资源文件添加	8
三、接入同步SDK平台	8
1、导入头文件	8
2、平台初始化	8
3、检查更新	10
4、平台登录、注销	12
1. 登录	12
2. 单设备应用之间自动登录说明	13
3. 判断玩家登录状态	13
4. 注销	13
5. 切换账号	13
6. 登录结果通知	13
7. 平台离开通知	15
5、虚拟币充值、物品购买	16
1、使用推币进行指定价格的商品购买	17
2、玩家自选金额进行虚拟币的充值兑换	20
3、测试支付充值	21
6、其他平台相关操作	21
1. 进入用户中心	21
2. 进入游戏推荐中心	21
3. 进入游戏论坛	21
四、同步游戏平台其他相关说明	21
1、升级规则（版本号设定规则）	21
Frequently Asked Question (FAQ)	23

开发前需知

在进行开发前，请先注意以下事宜：

1. 确保双方商务已完成合作事宜的洽谈，并签订合作协议；
2. 检查您是否已在同步商务创建的讨论组中，如果没有，请尽快联系商务将您加入讨论组中，以及时获得帮助；
3. 登录游戏联运中心 <http://dev.tongbu.com/game/> 获取 *Appid* 和 *Appkey* 。若应用列表为空，请添加应用，并联系我方商务进行审核；
4. 支付通知地址、代币单位名称、代币兑换比例和是否强制升级，均可在游戏联运中心进行配置；
5. 开发中遇到任何问题或不清楚的地方，请在讨论组中提出。

一、SDK构成

注：请到<http://dev.tongbu.com/game/>下载最新的SDK包。

同步移动平台SDK主要构成部分：

1. Framework

SDK从2.4版本开始，不再使用.a格式静态库，改为Framework包，架构支持armv7、armv7s，固件要求4.3（包含）以上版本。

2. 资源文件

SDK从2.4版本开始，为避免资源文件与开发者的资源文件或其他第三方资源文件命名冲突，改为使用bundle包。

3.1版本开始，SDK会检测资源包的版本，更新SDK时务必更换工程中的资源包。

3. Logo角标

接入同步游戏平台的App都需要打上同步平台的标志，详细参见同步游戏接入角标相关说明。

注：3.0版本以后的SDK，不再兼容2.4之前的旧接口，开发者注意使用新版接口进行接入。

同步SDK支持包括iPod Touch、iPhone、iPad在内采用armv7、armv7s和arm64架构的硬件设备，操作系统要求Mac OS X 10.7以上，Xcode要求4.2以上，iOS要求4.3以上。

（本手册相关示意图均在Xcode5.x环境下进行）

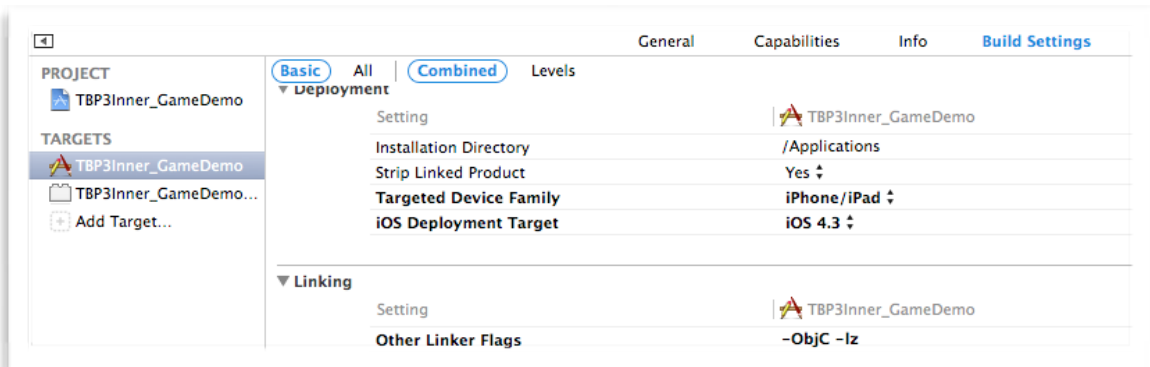
二、开发环境搭建

1、工程配置

* 打开工程配置，添加库的链接参数

在工程配置中，找到Linking中的Other Linker Flags，添加参数：

```
-ObjC -lz
```



注：如果项目之前有更改或使用的是Target中的配置，那么需要在Target的配置中做同样的修改。

2、添加Framework

1. 添加系统Framework和链接库

在工程Target中添加以下框架：

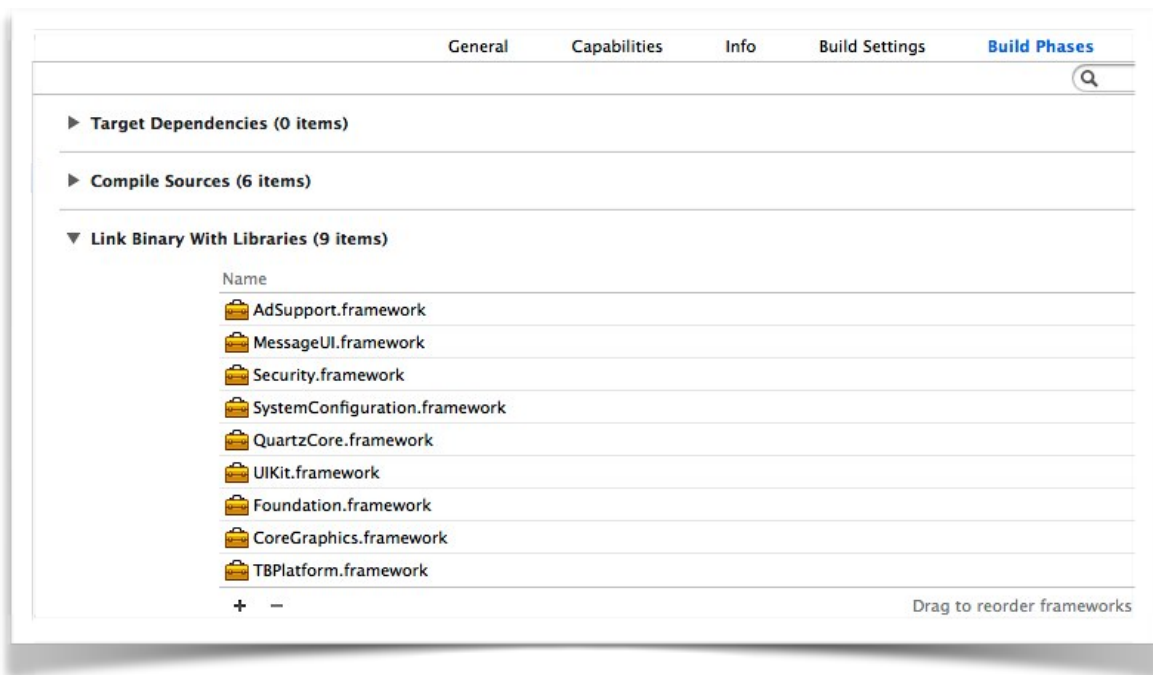
QuartzCore.framework

SystemConfiguration.framework

MessageUI.framework

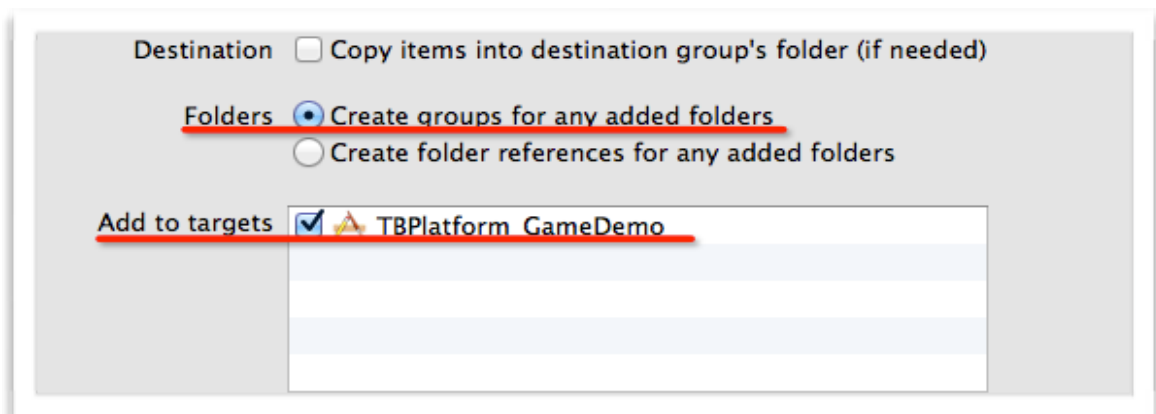
Security.framework

AdSupport.framework (3.1新增)

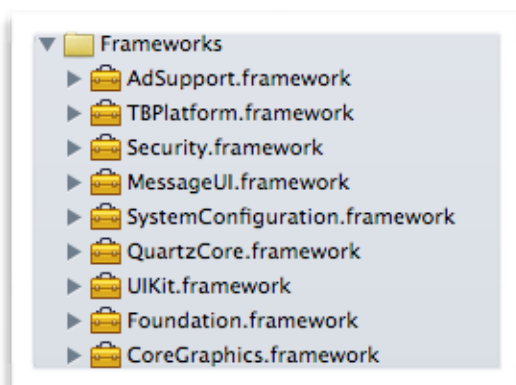


2. 添加同步SDK的Framework

右键工程中的Framework组，选择“Add Files to ”项目名称“...”，找到并选择TBPlatform.framework文件夹，注意采用“Create groups ...”方式添加，如下图：

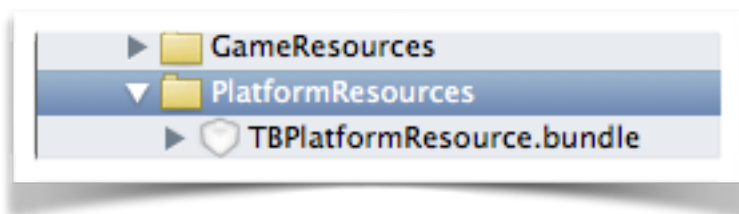


* 也可直接将Framework文件夹拖入工程，注意使用相同的创建选项
完成后Framework如下图：



3、资源文件添加

将SDK包中的资源文件TBPlatformResource.bundle使用“Add Files to ...”方式或拖入方式添加到工程中的相关资源目录下：



三、接入同步SDK平台

配置好开发环境，导入SDK框架和资源后，您就可以开始接入SDK的功能。

1、导入头文件

使用同步接入平台，需要导入SDK框架中的头文件，导入TBPlatform.h后会自动导入所有相关接口头文件：

```
#import <TBPlatform/TBPlatform.h>
```

头文件说明：

"TBPlatformBase.h"	平台的一些设置相关接口
"TBPlatform+Login.h"	登录相关接口
"TBPlatform+PayAndRecharge.h"	支付充值购买相关接口
"TBPlatform+Center.h"	进入各种中心接口
"TBNotifications.h"	各类通知定义
"TBPlatformAPIResponse.h"	一些数据类结构描述
"TBPlatformError.h"	错误码定义

2、平台初始化

AppID请到同步游戏联运后台管理系统中查看（需要先创建应用），如图：



图 3-2-1 获取应用基本信息



图 3-2-2 编译所用的SDK版本

注：如开发者使用iOS的SDK版本大于等于7.0（如图3-2-2），则需要打开iOS7特性支持开关：

```
[[TBPlatform defaultPlatform] TBSetSupportIOS7:YES];
```

为保证后续接入平台其他功能的正常工作，建议在游戏启动完毕后马上进行平台初始化工作（该方法会进行游戏版本检查工作）：

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions{
    self.window = [[[UIWindow alloc] initWithFrame:
        [[UIScreen mainScreen] bounds]] autorelease];
    [application setStatusBarHidden:YES];
    // Override point for customization after application launch.
    [self.window makeKeyAndVisible];
    /* 游戏开发者应在游戏启动时优先对平台进行 */
    [[TBPlatform defaultPlatform] TBInitPlatformWithAppID:100000
        screenOrientation:UIInterfaceOrientationLandscapeRight
        isContinueWhenCheckUpdateFailed:NO];
    return YES;
}
```

平台初始化工作完成后，会发送初始化成功通知，开发者应在该通知中继续相关工作（如登录）：

```
[[NSNotificationCenter defaultCenter] addObserver:self
                                     selector:@selector(tblInitDidFinish:)
                                     name:kTBInitDidFinishNotification
                                     object:nil];
```

```
-(void)tblInitDidFinish:(NSNotification *)notify{
    //初始化结果，即检查更新结果，如果初始化时允许在失败的情况下进入游戏，此时应该根据结果做相应处理
    int updateResult = [[notify.userInfo objectForKey:@"updateResult"]
intValue];
    //。。。
    [[TBPlatform defaultPlatform] TBLogin:0];
}
```

3、检查更新

平台初始化方法包含了游戏的检查更新工作，为了保证版本比较的准确性，需要严格按照标准定义版本号，具体参见版本号设定规则。

强制更新：玩家需要更新完才能继续游戏。

非强制更新：玩家不需要更新也可以进行游戏。

*是否强制更新请到联运后台设置，如图：

游戏管理 » 应用配置

支付通知地址:

http://localhost/result.html

接受支付结果通知的Url地址, 例如: http://localhost/result.html

游戏代币单位:

元宝

(默认"元宝")

代币兑换比例:

1元人民币 = 10

个代币 (默认"10")

是否强制升级:

否

✓ 是

开启后, 若用户的版本低于最新版, 将强制要求他升级。

提交

注: 确保游戏内配置的版本号与所计划发布的版本号一致, 每次升级需记得调整该处的版本号。

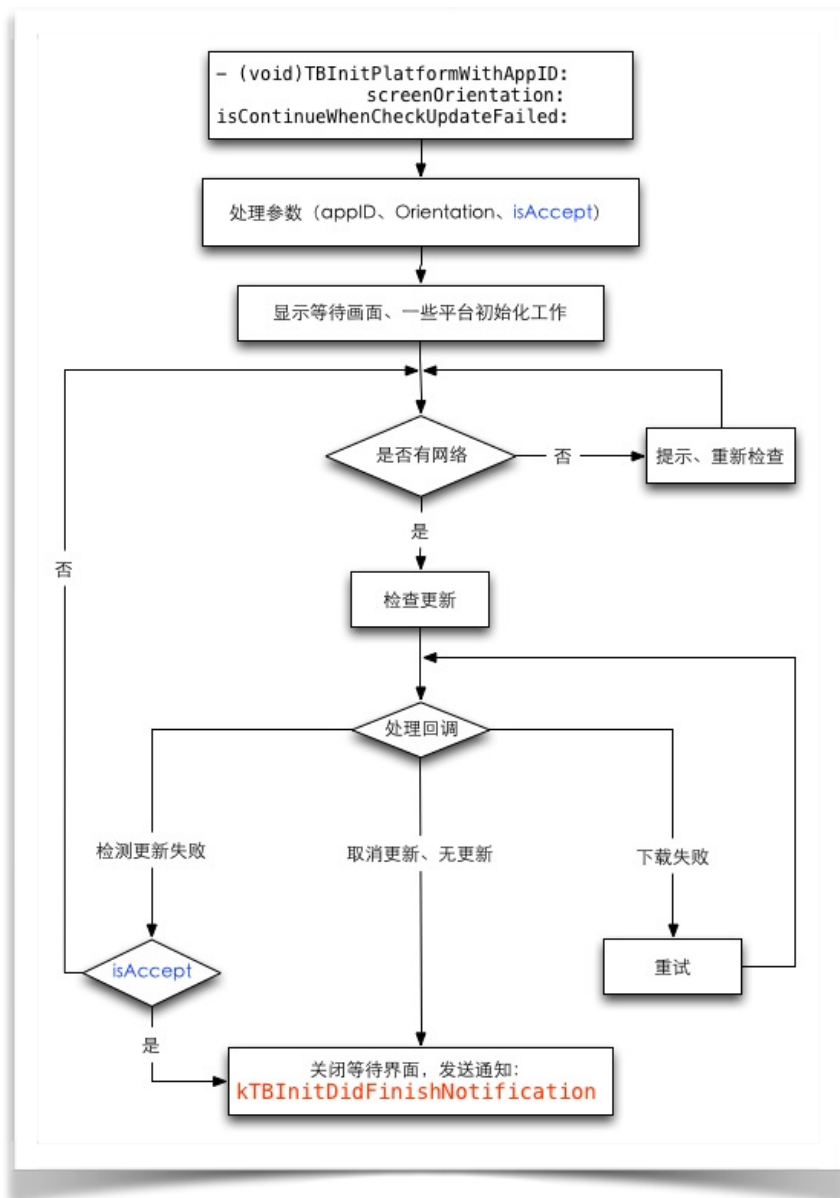
平台初始化接口将自动进行版本检查工作, 开发者在调用初始化方法时第三个参数为在检查失败时 (网络错误或服务器错误) 是否允许进入游戏, 如果允许进入游戏, 则在初始化结果通知里需要对检查失败的情况做处理, 否则可能导致强制更新可跳过等问题。

注意事项:

*** 在无网络的情况下, 不允许进入游戏。**

其他注意事项: 自动更新会从同步的下载服务器中获取游戏的完整安装包到本地并重新安装, 由此会导致iOS系统下的数据可能会丢失, 也就是说本地存储的部分游戏数据会丢失, 建议将需要保留的数据存放于Documents目录下, 并做重新安装后数据保留性测试, Documents目录在重新安装过程中, 会被系统拷贝一份到新安装拷贝中去, 但如果直接卸载游戏再重新安装, 这个目录的数据也将被清除。

自动检查更新大致流程图:



4、平台登录、注销

1. 登录

登录接口：

```
[[TBPlatform defaultPlatform] TBLogin:0];
```

说明：该接口在玩家有自动登录记录的情况下首次调用该接口将进行自动登录，无登录记录则进入登录界面，登录界面可选择注册新的账号等。开发者注意进行登录结果通知处理（见3.5.6）。

2. 单设备应用之间自动登录说明

3.0SDK开始，同步移动平台提供同步账号跨应用间自动登录，只要在该设备上登录过同步账号（在同步推、接入3.0SDK的其他游戏中），对于没有本地登录记录的游戏，会自动填入最近登录的账号及密码。即新下载的游戏可实现快速登录。

3. 判断玩家登录状态

判断玩家是否登录：

```
[[TBPlatform defaultPlatform] isLoggedIn];
```

4. 注销

注销可选择是否清空用户自动登录信息，0表示保留，1表示清除，默认为0：

```
[[TBPlatform defaultPlatform] TBLogout:0];
```

注销后平台会发送注销的通知：[kTBUserLogoutNotification](#)

开发者要注意玩家注销后及时更新游戏场景

5. 切换账号

切换账号接口即实现注销+登录：

```
[[TBPlatform defaultPlatform] TBSwitchAccount];
```

注：该接口调用后会发送注销通知，游戏需要接收该通知进行游戏场景处理。
（个人中心底部的切换账号按钮实现该接口）

6. 登录结果通知

玩家进行登录操作并且成功后，平台会发送登录结果通知，开发者需要在这个通知中进行相应的处理。注意在收到通知后再进行登录成功相关操作，以保证玩家当前的正确登录状态。

注意：只有在登录成功的情况下会发送该通知。

```
/* 监听登录结果通知 */  
[[NSNotificationCenter defaultCenter] addObserver:self  
                                         selector:@selector(loginResult:)  
                                         name:(NSString *)kTBLoginNotification  
                                         object:nil];
```

处理方法：

```
/**
 * @brief 登录通知监听方法，登录成功、失败都在这个方法处理
 *
 * @param notification 通知的userInfo包含登录结果信息
 */
- (void)loginResult:(NSNotification *)notification
{
    NSDictionary *dict = [notification userInfo];
    BOOL success = [[dict objectForKey:@"result"] boolValue];
    /*登录成功后处理*/
    if([[TBPlatform defaultPlatform] isLoggedIn] && success) {
        /*进行玩家信息的更新，或转入游戏界面等*/
        [self updatePlayerInfo];
        [self dismissModalViewControllerAnimated:YES];
    }
}
```

说明：登录通知结果返回的通知中带的userInfo内容。

```
/*[notification.userInfo allKeys]打印结果*/
(
    error,    /*相关错误代码*/
    tbuserinfo, /*TBPlatformUserInfo，用户信息*/
    result    /*登录结果，BOOL型*/
)
```

登录成功后可进行用户信息的获取（强烈建议使用userID作为游戏角色的唯一标识，nickName在特殊环境下可能存在重复的情况，使用nickName作为游戏角色唯一标识所产生的未知后果由游戏方自负）：

```
[[TBPlatform defaultPlatform] nickname]; /*用户昵称*/
[[TBPlatform defaultPlatform] sessionId]; /*用户会话ID*/
[[TBPlatform defaultPlatform] userID]; /*用户ID，唯一标识*/
```

* 登录成功后，可以根据sessionId，向服务器验证当前会话的有效性。

请求为GET请求。接口数据传输格式返回是文本格式，返回编码为UTF8。

检测Session是否有效接口说明（从开发者服务器或客户端发起验证都可）：

请求地址

<http://tgi.tongbu.com/checkv2.aspx>

接口描述：获取游戏联运Token有效性（时效2分钟）

参数格式：k=XXXX

示例： http://tgi.tongbu.com/checkv2.aspx?k=sessionid_string

返回数据格式

大于0：Token有效，具体数值为userID

0：失效

-1：格式有错

7. 平台离开通知

在关闭所有同步游戏平台SDK界面时都会发出该通知（登录成功后自动离开平台时不通知），其中的userInfo为离开的页面相关信息字典，其中的返回类型枚举：

```
typedef enum _TB_LeavedPlatform_Type_
{
    //离开未知平台（预留状态）
    TBPlatformLeavedDefault,
    //离开注册、登录页面（登录成功不发送离开通知）
    TBPlatformLeavedFromLogin,
    //包括个人中心、游戏推荐、论坛
    TBPlatformLeavedFromUserCenter,
    //离开充值页（包括成功、失败）
    TBPlatformLeavedFromUserPay,
}TBPlatformLeavedType;
```

开发者应接收该通知，根据各类情况做出游戏内处理：

```
/* 监听离开平台通知 */
[[NSNotificationCenter defaultCenter] addObserver:self
                                           selector:@selector(leavePlatform:)
                                           name:(NSString *)kTBLeavePlatformNotification
                                           object:nil];
```

处理方法：

```
/**
 *    @brief 平台关闭通知，监听离开平台通知以对游戏界面进行重新调整
 */
- (void)leavePlatform:(NSNotification *)notification
{
    TBPlatformLeavedType leavedType = [notification.userInfo
objectForKey:TBLeavedPlatformTypeKey];
    switch (leavedType) {
        //从登录界面手动关闭（登录成功关闭时不发送该通知）
        case TBPlatformLeavedFromLogin:
            if (![TBPlatform defaultPlatform] isLoggedIn){
                //回到登录界面，或其他相关操作...
            }
            break;
        //从个人中心页关闭（包括游戏推荐、论坛页）
        case TBPlatformLeavedFromUserCenter:
            break;
        //从充值页面退出（包含一个订单号字段）
        case TBPlatformLeavedFromUserPay:
            {
                NSString *order = [notification.userInfo
objectForKey:TBLeavedPlatformOrderKey];
                //查询订单结果或其他操作...
            }
            break;
        default:
            break;
    }
}
```

5、虚拟币充值、物品购买

3.0以上（包含3.0）版本SDK不支持未登录充值。

注意：开发者需要在后台配置相关服务器回调通知地址，如图：



游戏管理 » 应用配置

支付通知地址:

接受支付结果通知的Url地址，例如：http://localhost/result.html

SDK提供两个充值购买相关接口：

1、使用推币进行指定价格的商品购买

接口：

```
/**
 * @brief 进行虚拟币充值或商品购买（需登录，同步后台记录充值账号记录）
 * @param orderSerial 合作商订单号，必须保证唯一，双方对账的唯一标记
 * @param needPayRMB 需要支付的金额，单位：元
 * @param payDescription 支付描述，发送支付成功通知时，返回给开发者
 * @param delegate 回调对象，见TBBuyGoodsProtocol协议
 * @result 错误码
 */
- (int)TBUiPayForCoin:(NSString*)orderSerial
    needPayRMB:(int)needPayRMB
    payDescription:(NSString*)payDescription
    delegate:(id<TBBuyGoodsProtocol>)buyDelegate;
```

说明：该接口为指定价格的物品购买接口，传入needPayRMB为玩家所需要支付的人民币数量（1RMB=10推币）。购买分为两种情况：

一、在推币足以支付该订单时，提示用户进行直接购买，并通过回调对象通知购买结果，回调协议TBBuyGoodsProtocol如下：

```
/**
 * @brief 购买相关回调，通知购买结果
 */
@protocol TBBuyGoodsProtocol <NSObject>
typedef enum {
    kBuyGoodsBalanceNotEnough,    /*余额不足*/
    kBuyGoodsServerError,         /*服务器错误*/
    kBuyGoodsOrderEmpty,          /*订单号为空*/
    kBuyGoodsNetworkingError,     /*网络不流畅（有可能已经购买成功但客户端已超时，建议进行订单查询）*/
    kBuyGoodsOtherError,          /*其他错误*/
}TB_BUYGOODS_ERROR;
```

```
@optional
/**
 *    @brief 使用推币直接购买商品成功
 *    @param    order 订单号
 */
- (void)TBBuyGoodsDidSuccessWithOrder:(NSString*)order;
/**
 *    @brief 使用推币直接购买商品失败
 *    @param    order 订单号
 *    @param    errorType 错误类型，见TB_BUYGOODS_ERROR
 */
- (void)TBBuyGoodsDidFailedWithOrder:(NSString *)order resultCode:
(TB_BUYGOODS_ERROR)errorType;
/**
 *    @brief 推币余额不足，进入充值页面（开发者需要手动查询订单以获
取充值购买结果）
 *
 *    @param    order 订单号
 */
- (void)TBBuyGoodsDidStartRechargeWithOrder:(NSString*)order;
/**
 *    @brief 跳提示框时，用户取消
 *
 *    @param    order 订单号
 */
- (void)TBBuyGoodsDidCancelByUser:(NSString *)order;
@end
```

二、在推币不足以支付该订单时，提示用户进入Web充值页进充值购买，此时通过回调对象（TBBuyGoodsProtocol）告知客户端用户余额不足，已进入充值页面。在退出充值页后会需手动调用订单查询接口进行查询。查询接口：

```
/**
 @brief 查询订单支付是否成功
 @param strCooOrderSerial    支付订单号
 @param delegate    回调对象，回调接口参见 TBPayDelegate
 @result 错误码
 */
-(int)TBCheckPaySuccess:(NSString*)strCooOrderSerial
        delegate:(id)delegate;
```

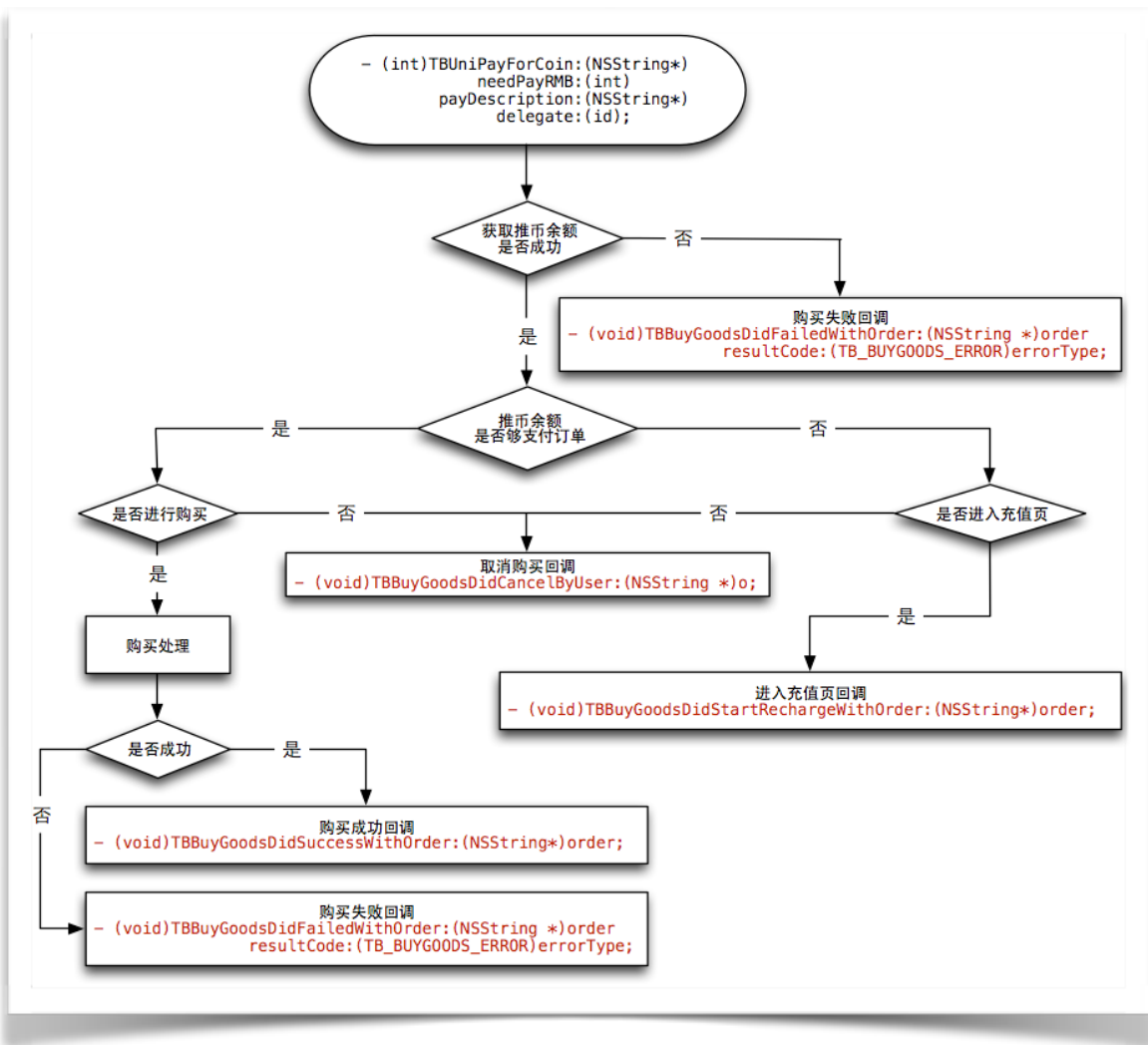
查询订单回调：

```
/**
 * @brief 手动查询充值结果回调协议
 */
@protocol TBPAYDelegate <NSObject>

/**
 * @brief 查询充值结果
 *
 * @param dict 结果信息
 *          order:订单号
 *          amount:充值金额（单位：分）
 *          status:状态（-1：其它错误（包括选定金额后未进入第
三方充值页或服务器返回异常）
                                0：待支付（已经创建第三方充值订单，
但未支付）
                                1：充值中（用户支付成功，正在通知开
发者服务器，未收到处理结果）
                                2：失败
                                3：成功（通知开发者服务器并收到处理
成功结果，充值完毕））
 */
- (void)TBCheckOrderSuccessWithResult:(NSDictionary *)dict;
/**
 * @brief 查询订单失败（网络不畅通，或服务器返回错误）
 */
- (void)TBCheckOrderDidFailed:(NSString*)order;

@end
```

支付流程图：



2、玩家自选金额进行虚拟币的充值兑换

玩家自选金额充值接口：

```

/**
 * @brief 进行充值。该接口直接进入Web页充值，无回调，开发者可以使用
 *         TBCheckPaySuccess:delegate:接口进行订单查询
 * @param orderSerial  合作商订单号，必须保证唯一，双方对账的唯一标记
 * @param payDescription 支付描述，发送支付成功通知时，返回给开发者
 * @result 错误码
 */
- (int)TBUniPayForCoin:(NSString*)orderSerial
    payDescription:(NSString*)payDescription;
  
```

调用该接口后直接进入Web充值页，由玩家选择需要充值的数量。建议开发者在退出离开平台的消息处理方法中调用TBCheckPaySuccess:delegate:接口进行

订单查询。

3、测试支付充值

请联系同步运营专员获取测试账号，测试账号中的推币只允许在对应的游戏中进行充值测试，测试账号生成的所有订单不纳入结算。

6、其他平台相关操作

1. 进入用户中心

针对使用同步账号的用户， 用户可以进行查看当前账号，修改密码，查看充值记录，切换账号，查看帮助等操作。

```
/* 若用户未登录，则自动进入登录界面 */  
[[TBPlatform defaultPlatform] TBEnerUserCenter:0];
```

2. 进入游戏推荐中心

进入同步平台的游戏推荐中心，用户可以查看同步平台近期推荐的游戏。

```
/* 若用户未登录，则自动进入登录界面 */  
[[TBPlatform defaultPlatform] TBEnerAppCenter:0];
```

3. 进入游戏论坛

(注：默认论坛不开启，如果有论坛，请联系同步运营专员)

```
int bbsResult = [[TBPlatform defaultPlatform] TBEnerAppBBS:0];  
/* 需要判断是否开启论坛 */  
if (bbsResult == TB_PLATFORM_NO_BBS) {  
    [self showMessage:@"该游戏未配置论坛"];  
}
```

四、同步游戏平台其他相关说明

1、升级规则（版本号设定规则）

为了进行版本比较和升级，需要统一版本格式。

统一格式：A.B.C.D (其中A、B、C、D为整数)

另外允许 **A.B** 等同于 **A.B.0.0**，**A.B.C** 等同于 **A.B.C.0** 比较优先级次序，优先比较A，其次比较B，再次C，最后D。数字大的为高版本。

例如：

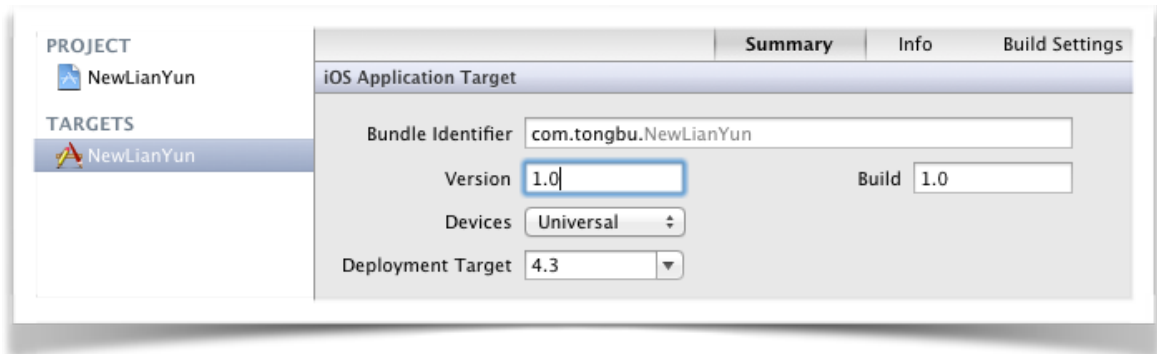
1.20 大于 1.2.1

2.0 大于 1.100

1.05 等于 1.5

设定说明:

在Info.plist中, 有两个版本号相关的字段, 分别是CFBundleShortVersionString和CFBundleVersion。其中的CFBundleShortVersionString对应图中的Version, 而CFBundleVersion 对应的是Build。



平台版本升级时:

- 查找CFBundleIdentifier(Bundle Identifier)字段。
- 使用CFBundleShortVersionString(Version)字段值进行比较。
- 在没有CFBundleShortVersionString(Version)字段的情况下, 才会将CFBundleVersion(Build)字段作为版本号进行比较。
- 如果没有其他的特殊情况而同时设定两个字段时, 建议开发者将这两个字段设定成相同的版本号。

Frequently Asked Question (FAQ)

1、为什么iOS6下进入SDK，再出来输入法不能输入？

确认SDK是最新版，如果确定是最新版，再确认是不是在调用了[window makeKeyAndVisible];之后进行SDK平台相关调用。

2、为什么明明充值了，却没有收到通知？

确认appid和key是正确的，确认回调url已经配置并正确。

3、出现下图中的错误

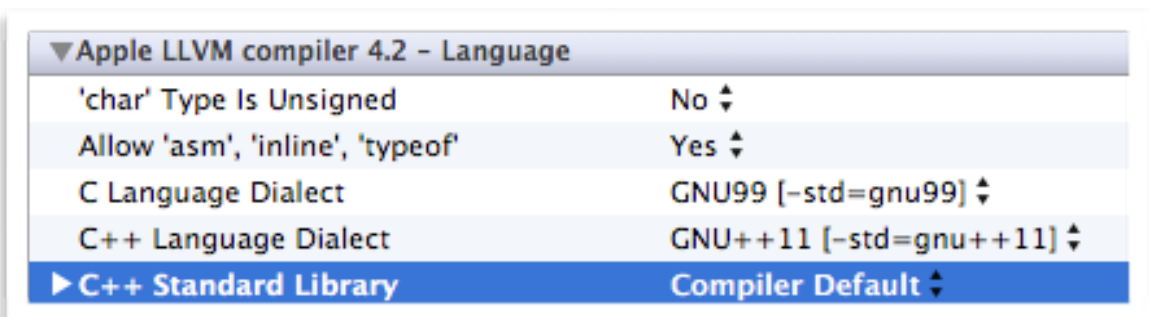
```
Undefined symbols for architecture i386:
  "operator delete(void*)", referenced from:
    encrypt(unsigned char*, unsigned char*, int, DESMode) in TBFramework(TBDESVariant.o)
  "operator new(unsigned long)", referenced from:
    encrypt(unsigned char*, unsigned char*, int, DESMode) in TBFramework(TBDESVariant.o)
ld: symbol(s) not found for architecture i386
clang: error: linker command failed with exit code 1 (use -v to see invocation)

❗ "operator delete(void*)", referenced from:
  Encrypt(unsigned char*, unsigned char*, int, DESMode) in TBFramework(TBDESVariant.o)
❗ "operator new(unsigned long)", referenced from:
  Encrypt(unsigned char*, unsigned char*, int, DESMode) in TBFramework(TBDESVariant.o)
Symbol(s) not found for architecture i386
❗ Linker command failed with exit code 1 (use -v to see invocation)
```

最简单的方法是将任意一个.m的文件改为.mm 保证能正确编译C++相关内容

4、编译出现错误：clang: error: invalid deployment target for -stdlib=libc++ (requires iOS 5.0 or later)


修改最低发行版本到5.0或修改C++库为Compiler Default，如图：



5、出现下图错误：


```
Undefined symbols for architecture armv7:  
  "std::ios_base::~Init::~~Init()", referenced from:  
    __GLOBAL__I_a in libUPPayPlugin.a(UPPasswordTool.o)  
  "std::ios_base::~Init::~Init()", referenced from:  
    __GLOBAL__I_a in libUPPayPlugin.a(UPPasswordTool.o)  
ld: symbol(s) not found for architecture armv7  
clang: error: linker command failed with exit code 1 (use -v to
```

```
! "std::ios_base::~Init::~~Init()", referenced from:  
  __GLOBAL__I_a in libUPPayPlugin.a(UPPasswordTool.o)  
! "std::ios_base::~Init::~Init()", referenced from:  
  __GLOBAL__I_a in libUPPayPlugin.a(UPPasswordTool.o)  
Symbol(s) not found for architecture armv7  
! Linker command failed with exit code 1 (use -v to see invocation)
```

 Activity Log Complete 13-2-28 下午2:38
3 errors, 8 warnings

该错误为加入银联库导致，将XCode工程中Build Setting -> Apple LLVM compiler Language 标签下的C++ Language Dialect 和 C++ Standard Library 属性值修改为 Compiler Default 即可。