

多盟激励视频－SDK接入文档（iOS）

1. 必要条件

- 使用Xcode 7或更高版本
- 运行环境为iOS 8.0或更高版本

2. SDK简述

- 文件方面，SDK包含一个静态库 `libIndependentVideoSDK.a`、一个头文件 `IndependentVideoManager.h`、一个资源文件包 Resource 文件夹 (`IndependentVideoBundle.bundle`, `en.lproj` , `zh-Hans.lproj`)。
- API方面，SDK提供 `IndependentVideoManager` 对象，用于展现视频界面

3. 视频SDK的使用

3.1 获取Publisher ID

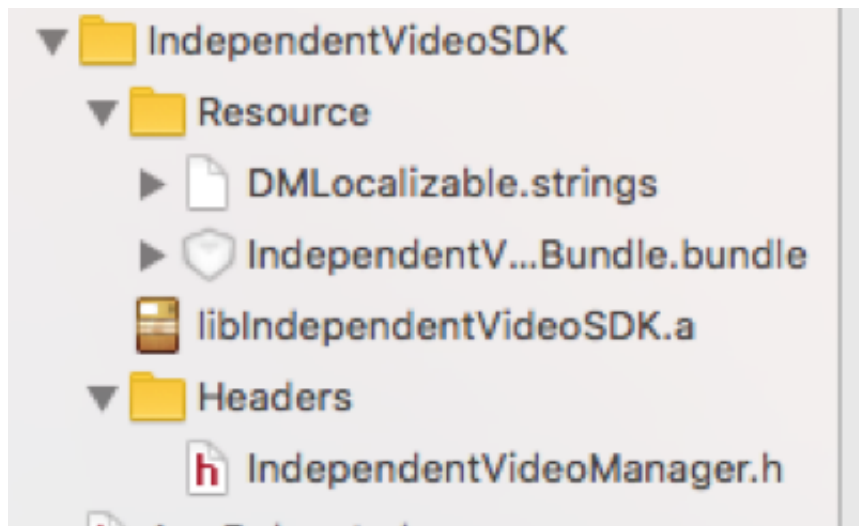
在多盟开发者后台(<http://www.domob.cn>)添加媒体获得对应的Publisher ID。

3.2 集成SDK的准备工作

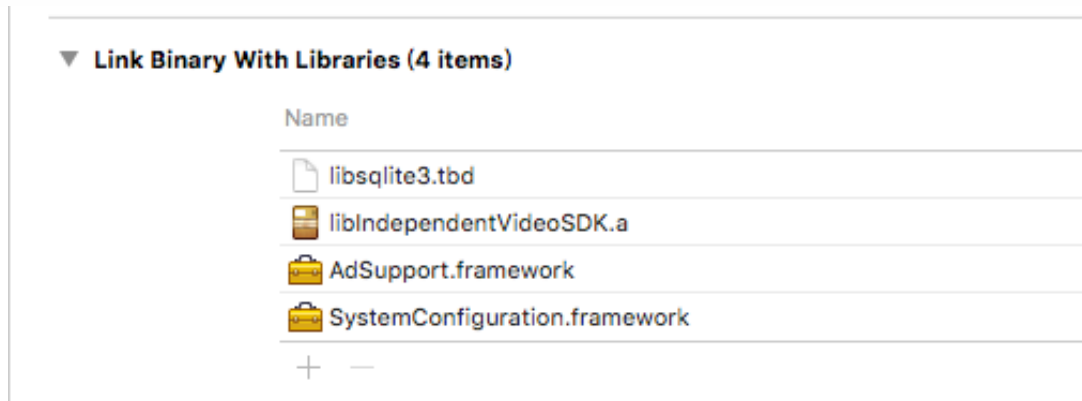
视频SDK的发布版本中需要添加到项目中的文件包括一个静态库 `libIndependentVideoSDK.a`、一个头文件 `IndependentVideoManager.h` 以及一个资源文件包Resource文件夹。

3.2.1 将上述文件，添加到您的项目中。建议直接将“IndependentVideoSDK”文件夹添加到项目中。

包含的内容，如下图：

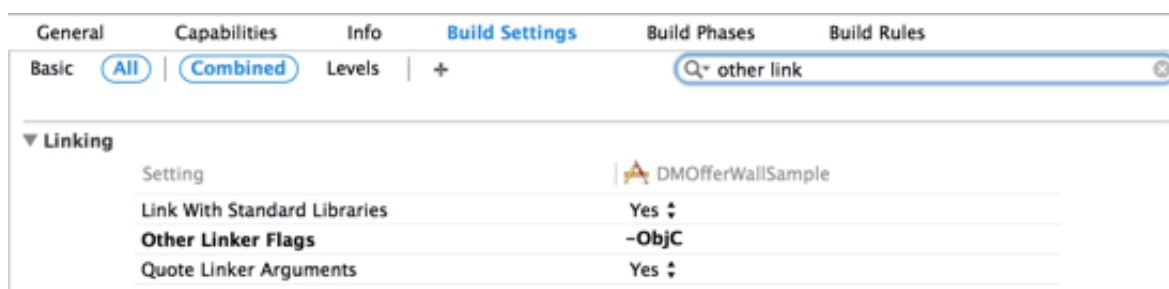


3.2.2 若要SDK正常工作，并添加一些库，如下图：



如有必要，再添加其他framework

3.2.3 BuildSettings中other link flags 选项添加-ObjC，如下图：



3.3 最快速集成SDK

3.3.1 初始化视频SDK后调用 `checkVideoAvailable` 方法，检查并下载视频

```
// 初始化
_videoManager = [[IndependentVideoManager alloc]
initWithPublisherID:self.publisherID andUserID:self.userID];
// 设置代理对象
_videoManager.delegate = self;
// 调用check方法
[_videoManager checkVideoAvailable];
```

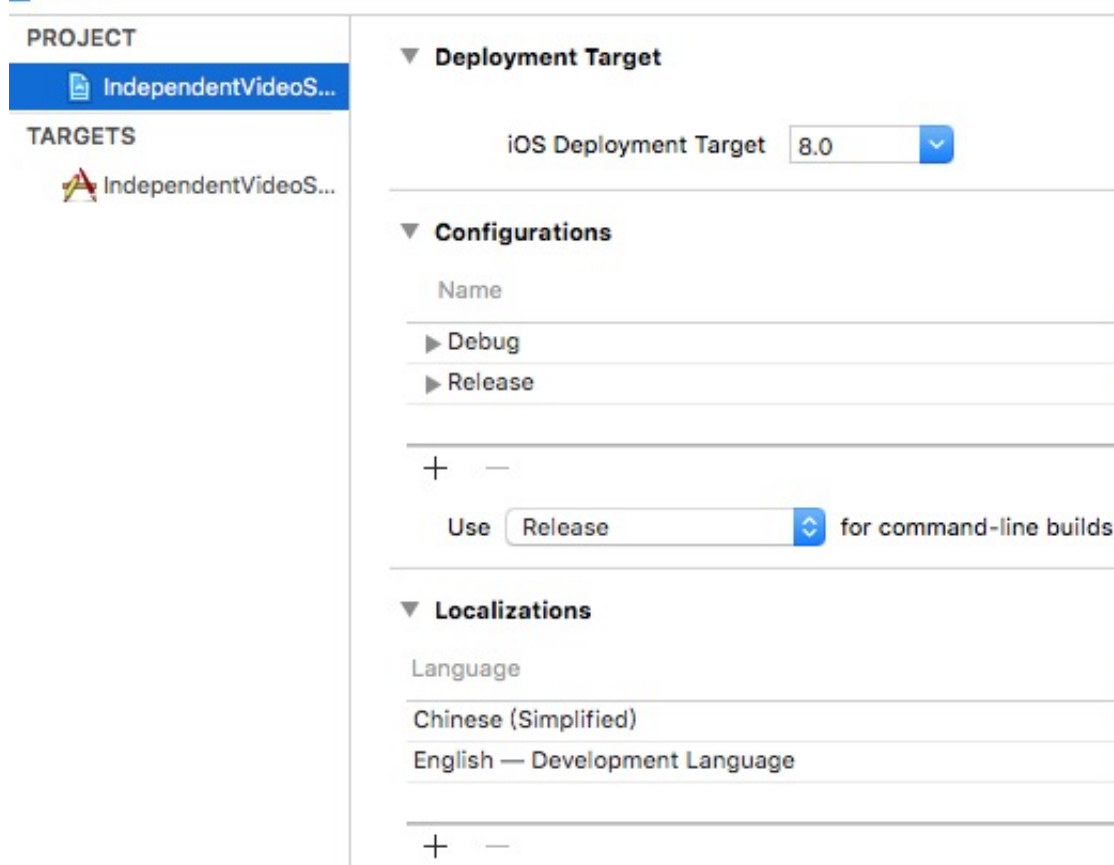
3.3.2 如果当前有视频，则调用回调三方法， `isFinished` 有YES/NO值

```
// 有视频时,加载是否完成。
- (void)ivManagerDidFinishLoad:(IndependentVideoManager *)manager finished:
(BOOL)isFinished {
    if (isFinished) {
        // 表示视频已缓存完毕，可以显示播放按钮
        _playButton.hidden = NO;
    }
}
}
注意:播放前先调用 checkVideoAvailable 方法,检查视频状态; 在isFinished=YES时再进行播放。
```

3.4 支持HTTPS

3.5 更新国际化(中文/英语)

1. 导入IndependentVideoSDK/Resource文件夹；
2. 内部有en.lproj 和zh-Hans.lproj文件，分别代表英语和中文两种国际化语言；
3. 在项目中添加支持相应的语言，PROJECT -- Info -- Localizations -- **+**号 -- 添加语言文件，如下图所示：
4. SDK即可根据iPhone系统当前使用的语言自动进行国际化适配。



3.6 视频SDK中API的介绍及使用

3.6.1 初始化视频SDK

```
#import "IndependentVideoManager.h"
@interface ViewController: UIViewController<IndependentVideoManagerDelegate>
// 视频SDK管理对象
@property (nonatomic, strong) IndependentVideoManager *_manager;
@end
```

```

@implementation ViewController
- (void)viewDidLoad {
    [super viewDidLoad];
    // 初始化SDK
    _manager = [[IndependentVideoManager alloc] initWithPublisherID:@"申请的publishId" andUserID:nil];
    // 设置代理对象
    _manager.delegate = self;
    // 检查视频状态
    [_manager checkVideoAvailable];
}

```

3.6.2 视频SDK代理回调方法

- 可以通过实现 `IndependentVideoManagerDelegate` 中定义的方法，来跟踪视频生命周期的各个阶段。
- 所有这些方法也都定义在 `IndependentVideoManager.h` 中，均是可选实现的，具体如下：

```

#pragma mark - video present callback 视频生命周期各个阶段的回调(八个回调方法)
@optional
/**
 * 回调一：
 * 调用checkVideoAvailable的回调方法
 * check independent video available.
 * available YES/NO 表示当前请求,服务端是否有视频广告数据返回。返回YES,会自动下载当前视频。
 * @param IndependentVideoManager
 * @param available
 */
- (void)ivManager:(IndependentVideoManager *)manager isIndependentVideoAvailable:(BOOL)available;

/**
 * 回调二：
 * 开始加载视频数据。
 * Independent video starts to fetch ad info.
 * @param manager IndependentVideoManager
 */
- (void)ivManagerDidStartLoad:(IndependentVideoManager *)manager;

/**
 * 回调三：
 * 加载是否完成,带参数回调。
 * Fetching independent video .
 * isFinished YES/NO 代表下载是否完成
 * @param manager IndependentVideoManager
 */

```

```

- (void)ivManagerDidFinishLoad:(IndependentVideoManager *)manager finished:
(BOOL)isFinished;

/**
 * 回调四：
 * 加载失败。可能的原因由error部分提供，例如网络连接失败、被禁用等。
 * Failed to load independent video.
 * @param manager IndependentVideoManager
 * @param error error
 */
- (void)ivManager:(IndependentVideoManager *)manager failedLoadWithError:(NSError
*)error;

/**
 * 回调五：
 * 视频界面被呈现出来时，回调该方法。
 * Called when independent video will be presented.
 * @param manager IndependentVideoManager
 */
- (void)ivManagerWillPresent:(IndependentVideoManager *)manager;

/**
 * 回调六：
 * 页面关闭。
 * Independent video closed.
 * @param manager IndependentVideoManager
 */
- (void)ivManagerDidClosed:(IndependentVideoManager *)manager;

/**
 * 回调七：
 * 当视频播放完成后，回调该方法。
 * Independent video complete play
 * @param manager IndependentVideoManager
 */
- (void)ivManagerCompletePlayVideo:(IndependentVideoManager *)manager;

/**
 * 回调八：
 * 播放视频失败
 * Paly independent video failed.
 * @param manager IndependentVideoManager
 * @param error
 */
- (void)ivManagerPlayIndependentVideo:(IndependentVideoManager *)manager withError:
(NSError *)error;

```

3.6.3 其他相关属性及方法

- 视频广告的视图会通过弹出的方式显示在界面的顶层;
- 检查是否有视频广告可以播放

```
// 检查视频方法
- (IBAction)checkVideoAvailable {
    [_manager checkVideoAvailable];
}
```

- 设置是否在播放完成后弹出弹框,默认为不弹出,为YES

```
@property (nonatomic, assign) BOOL disableShowAlert;
```

- 是否开启打印日志log,默认不开启,为NO

```
@property (nonatomic, assign) BOOL openLogger;
```

- 视频的显示:

```
- (void)showWithModal:(id)sender {
    [_manager presentIndependentVideo];
}
- (void)showWithModalWithViewController:(id)sender {
    [_manager presentIndependentVideoWithViewController:(UIViewController
*)controller];
}
```

3.6.4 设置User Id

- 如果不需要,可不进行设置,传nil;
- 当前用户生命周期内的激活都会记录在这个用户上,直到用户退出并登陆其它用户时,需要销毁之前的视频SDK实例并重新创建新的实例,带入新的用户名(User ID)。

注: userid需符合正则表达式[0-9a-zA-Z@._-]{1,64}

4. 其他

更多接口信息,请参考SDK包中的头文件中包含的信息。

更详细的使用方法,请参考发行文件中的Sample程序。