
科大讯飞股份有限公司

iFLYTEK CO.,LTD

科大讯飞人脸识别新手指南（iOS）

目录

1	概述	1
2	MSC 概述	1
2.1	iOS 人脸识别 SDK 接口和体系结构.....	1
3	搭建开发环境	2
3.1	创建 iOS 工程.....	2
3.2	添加静态库.....	2
3.3	添加 FRAMEWORK.....	5
3.4	确认 SDK 的路径	6
3.5	导入头文件.....	6
3.6	集成帮助文档到 XCODE	6
3.7	其他工程配置.....	9
4	在代码中使用开发工具包	9
4.1	人脸识别示例代码	9
4.1.1	人脸注册.....	10
4.1.2	人脸验证.....	11
4.1.3	人脸检测.....	12
4.1.4	人脸聚焦.....	12
4.2	详细 DEMO	12
4.3	使用注意.....	12
5	Q&A.....	12
6	附录.....	12
6.1	人脸识别结果说明.....	12
6.2	错误码列表.....	16

1 概述

本文档是开发科大讯飞 iOS 人脸识别 SDK 的用户指南，定义了人脸识别相关接口的使用说明和体系结构，所有接口必需在联网状态下才能正常使用。

其适用的读者为使用人脸识别 SDK 进行开发的产品设计师、软件工程师，通过阅读本文档，读者可以掌握如何集成和使用人脸识别服务。

2 MSC 概述

2.1 iOS 人脸识别 SDK 接口和体系结构

本文档是开发科大讯飞 iOS 人脸识别程序的用户指南，定义了人脸识别相关接口的使用说明和体系结构，如图 3-1 所示。

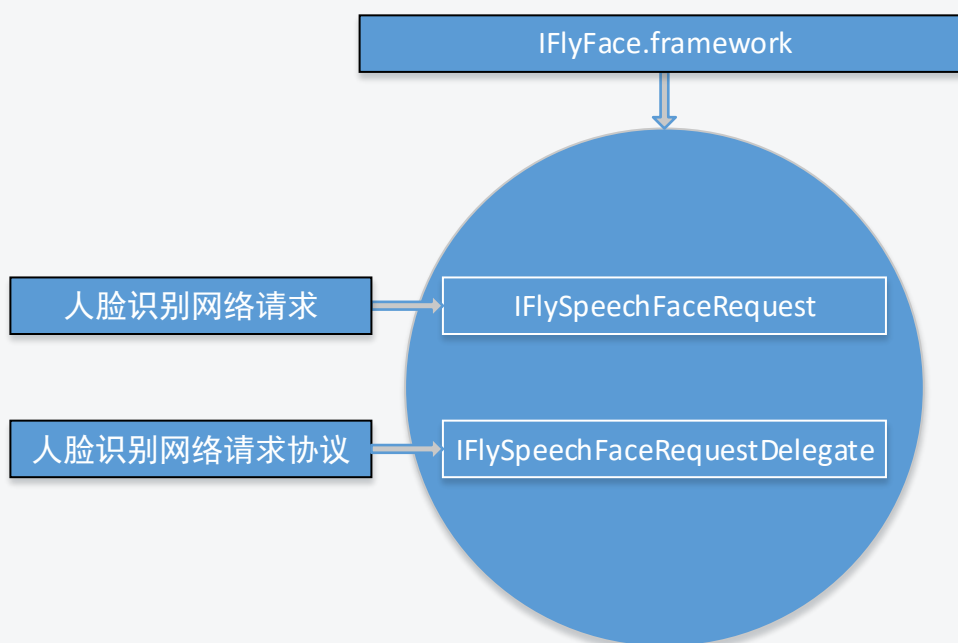


图 3-1 IFlyFace.framework 功能结构图

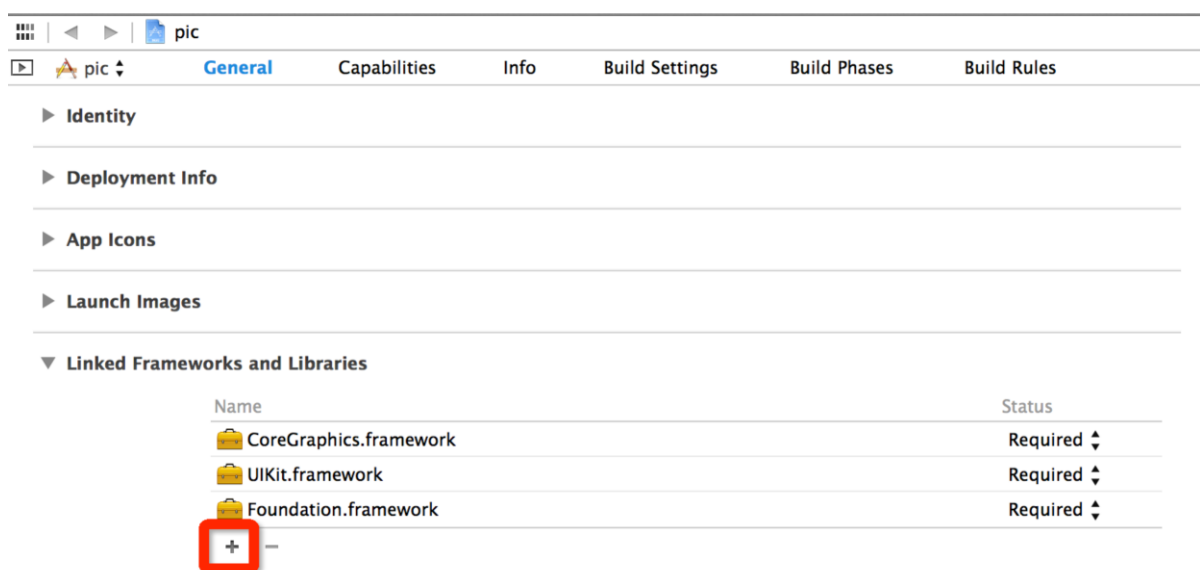
3 搭建开发环境

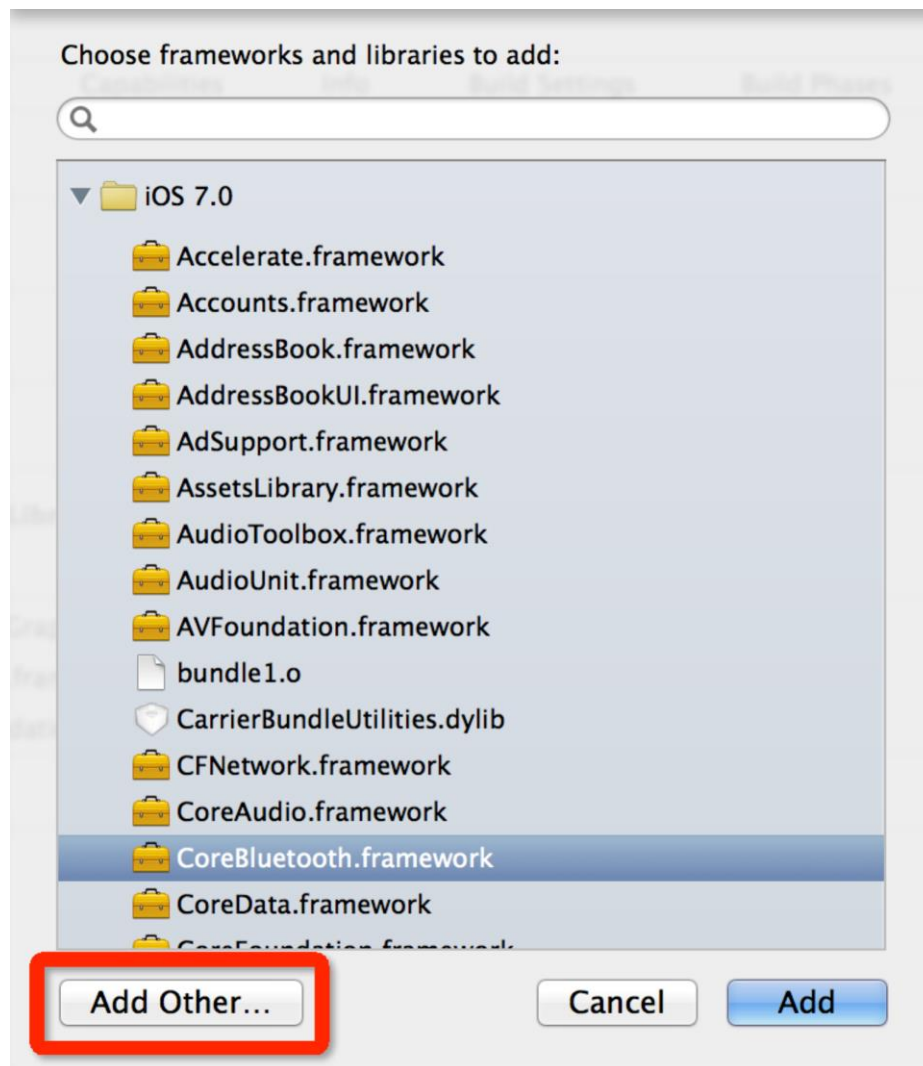
3.1 创建 iOS 工程

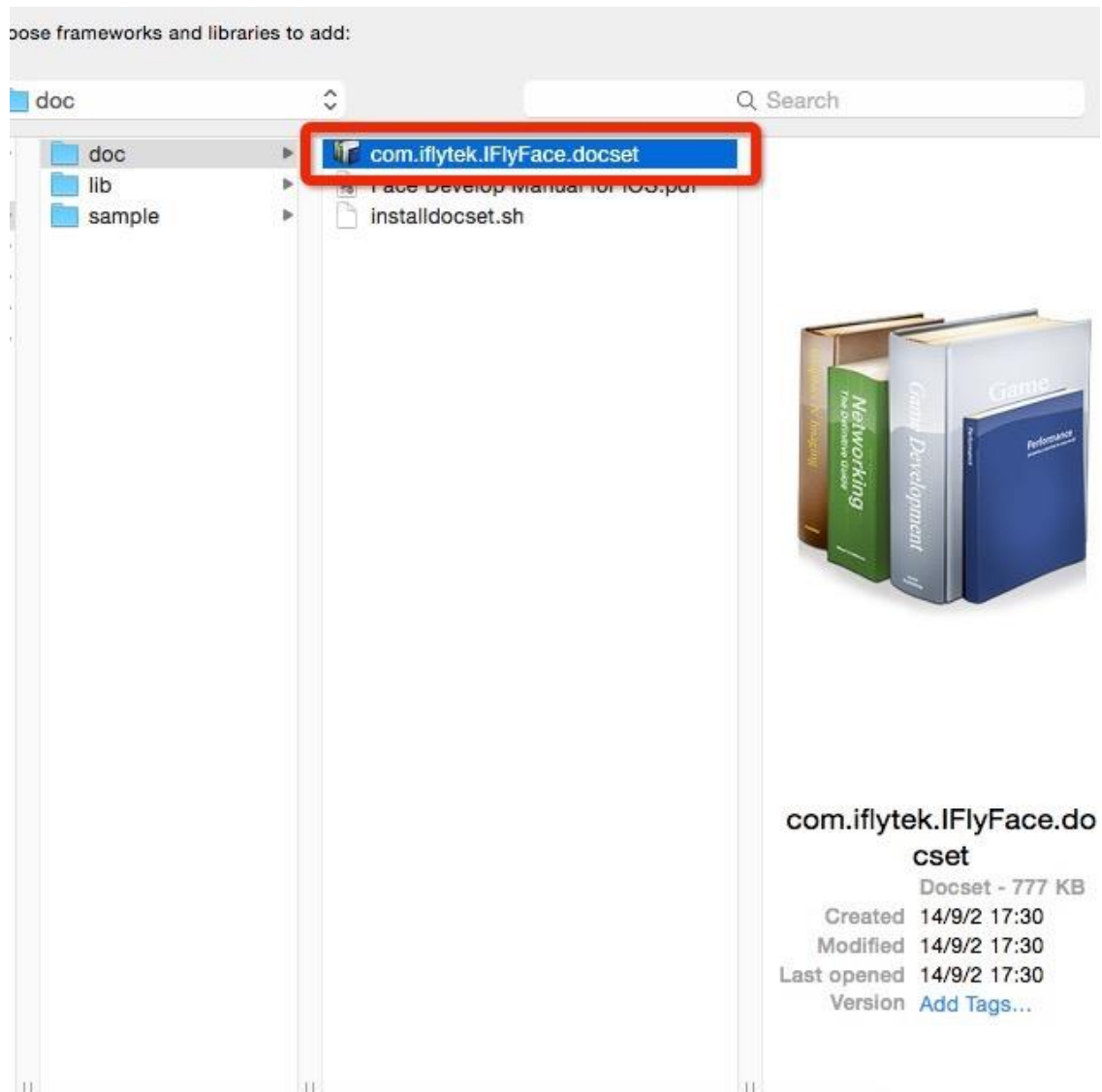
在 XCode 中建立你的工程，或者打开已经建立的工程。

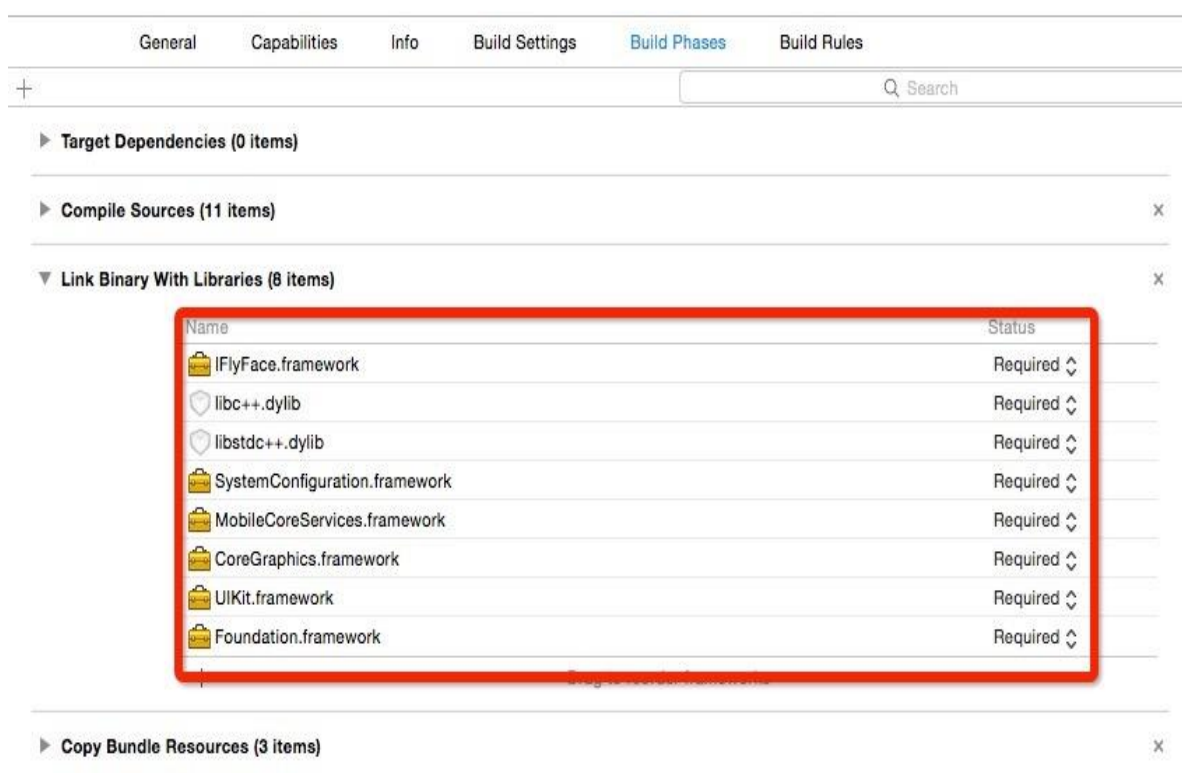
3.2 添加静态库

将开发工具包中 lib 目录下的 IFlyFace.framework 添加到新建工程中（如下图所示）。



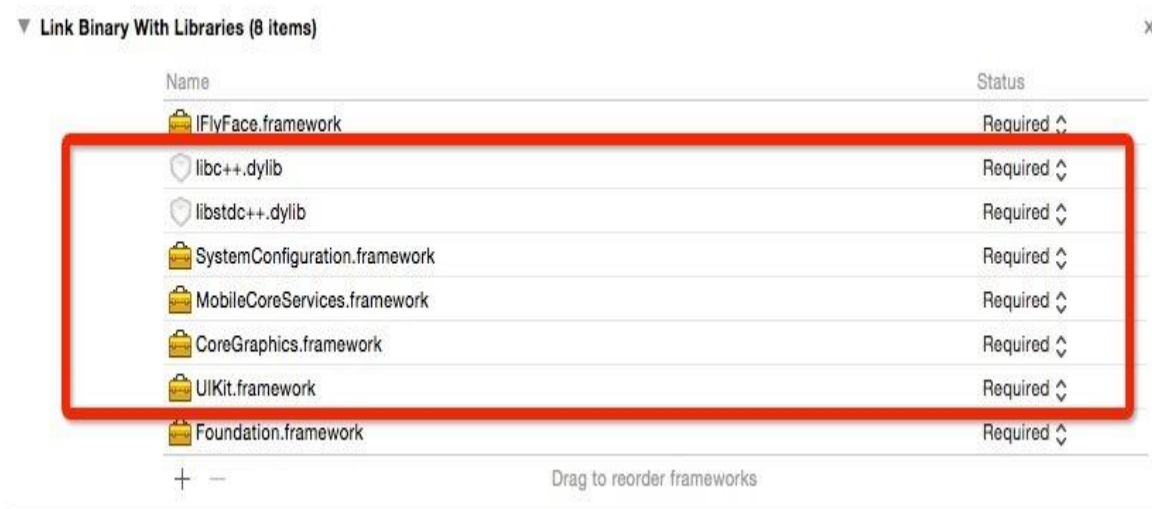




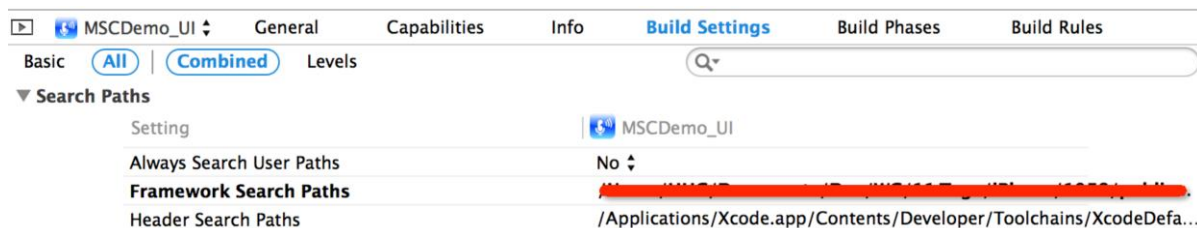


3.3 添加 framework

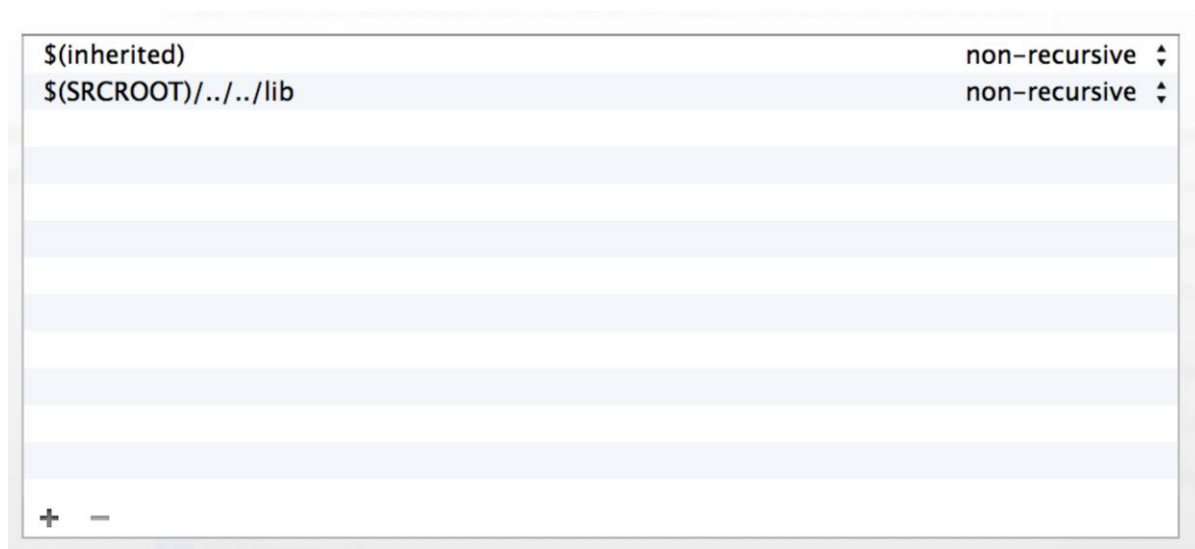
按下图添加 SDK 开发人脸识别应用时可能需要的系统 iOS 库



3.4 确认 SDK 的路径



请确认上图红色部分的路径能够找到 IFlyFace.framework。为了支持多人开发，建议双击红色部分，把路径改为相对路径，例如像下图所示。



注意：请把不必要的路径删除。例如更新了 SDK 后，新的 SDK 与旧的 SDK 不在同一路径，请把旧的路径删除，避免引用到旧的库。对应集成 SDK 后发现编译失败，提示找不到头文件，请先检查这个路径是否正确。

3.5 导入头文件

在你需要使用人脸识别服务的文件中导入相应的头文件

例如：

```
#import "IFlyFace.framework/IFlyFaceSDK.h"
```

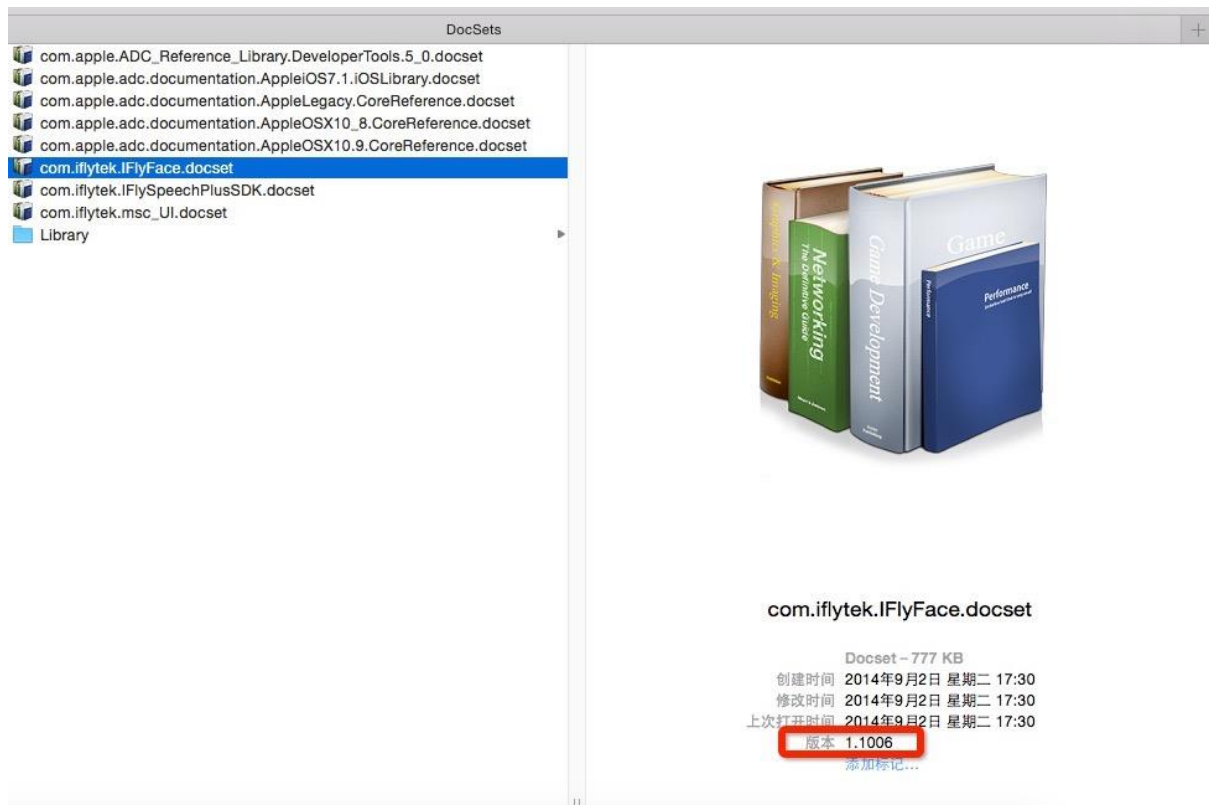
3.6 集成帮助文档到 Xcode

Xcode6 以下版本可以打开终端（terminal 或 iterm），cd 到压缩包的 doc 目录，执行以下命令：

```
cp -R -f -a com.iflytek.IFlyFace.docset ~/Library/Developer/Shared/Documentation/DocSets/
```


然后执行命令

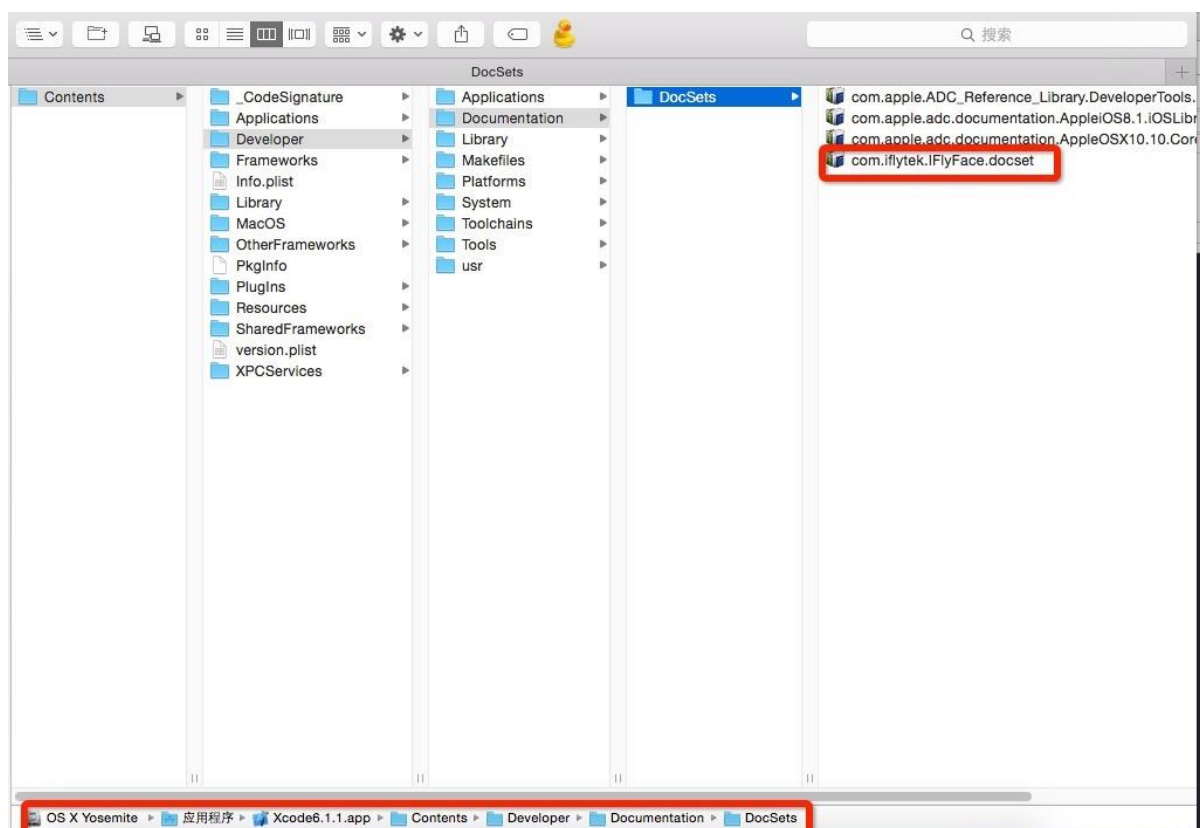
```
open ~/Library/Developer/Shared/Documentation/DocSets/
```



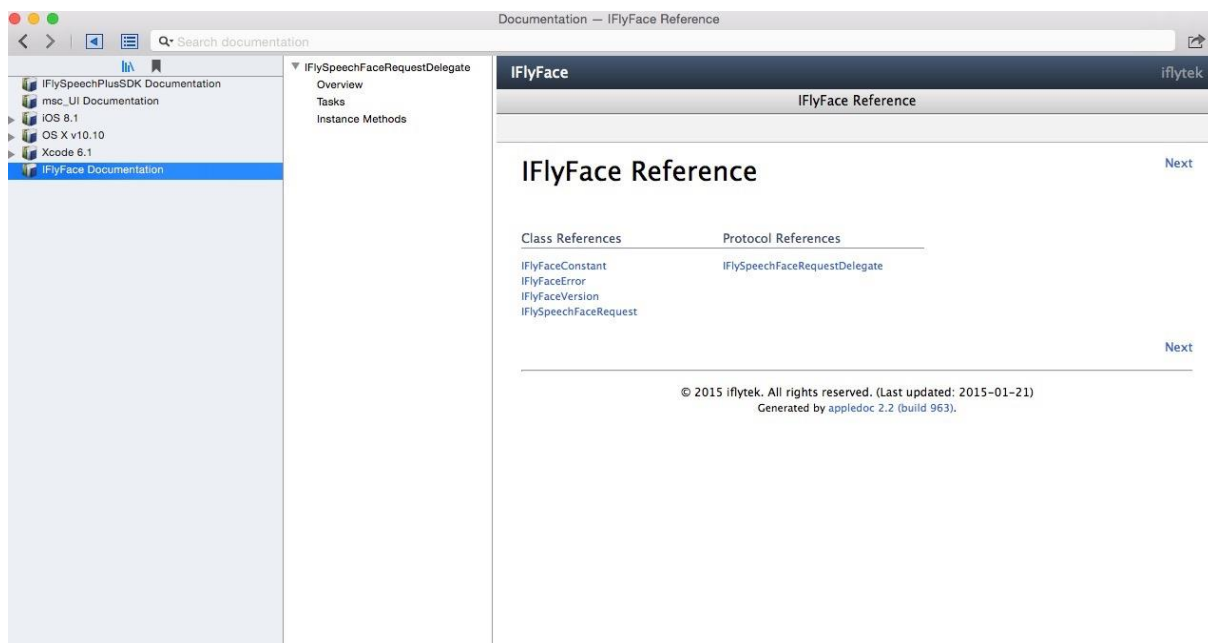
请核对文档的版本为最新下载的版本。

Xcode6 及以上版本以 (Xcode6.1.1 为例) 将 com.iflytek.IFlyFace.docset 拷贝到:

/Applications/Xcode6.1.1.app/Contents/Developer/Documentation/DocSets/



打开 Xcode 的帮助文档就可以看到已经集成的文档



3.7 其他工程配置

在 iOS 9 下直接进行 HTTP 请求时会收到如下错误提示：

App Transport Security has blocked a cleartext HTTP (http://) resource load since it is insecure.

Temporary exceptions can be configured via your app's Info.plist file.

不能直接使用 HTTP 进行请求，需要在 Info.plist 新增一段用于控制 ATS 的配置：

```
<key>NSAppTransportSecurity</key>
<dict>
    <key>NSAllowsArbitraryLoads</key>
    <true/>
</dict>
```

即：

▼ NSAppTransportSecurity	Dictionary	(1 item)
NSAllowsArbitraryLoads	Boolean	YES

4 在代码中使用开发工具包

4.1 人脸识别示例代码

人脸识别不仅可以检测出照片中的人脸，还可以进行人脸注册和验证。相关概念的说明如下：

表 4-1 人脸识别概念说明

名称	含义	说明
reg	注册	上传包含一张人脸的图片到云端，引擎对其进行特征提取，生成一个与之对应的模型，返回模型 id(gid)。
verify	验证	注册成功后，上传包含一张人脸的图片到云端，引擎将其特征与所注册的人脸模型进行比对，验证是否为同一个人，返回验证结果。
detect	检测	上传一张图片，返回该图片中人脸的位置（支持多张人脸）。
align	聚焦	上传一张图片，返回该图片中人脸的关键点坐标（支持多张人脸）。
gid	模型 id	人脸模型的唯一标识，长度为 32 个字符，注册成功后生成。一个用户（auth_id）理论上可以注册多个人脸模型（对应 gid），在进行验证时必须指定 gid。
auth_id	用户 id	由应用传入，用于标识用户身份，长度为 6-18 个字符（由英文字母、数字、下划线组成，不能以数字开头）。同一个 auth_id 暂时仅允许注册一个模型。
Property	属性	支持 del 和 undel，用于注册时删除/不删除原模型。

为了获得较高的准确率，请确保输入的图片满足以下要求：

表 4-2 上传图片规格要求

项目	要求
色彩、格式	彩色，PNG、JPG、BMP 格式的图片。

人脸大小、角度	大小应超过 100*100 像素，可以容忍一定程度的侧脸，为保证识别准确率，最好使用正脸图片。
光照	均匀光照，可容忍部分阴影。
遮挡物	脸部尽量无遮挡，眼镜等物品会一定程度上影响准确率。

4.1.1 人脸注册

```
// 创建人脸对象
self.iFlySpFaceRequest=[IFlySpeechFaceRequest sharedInstance];
[self.iFlySpFaceRequest setDelegate:self];
// 设置注册参数
[self.iFlySpFaceRequest setParameter:[IFlyFaceConstant REG] forKey:[IFlyFaceConstant SST]];
[self.iFlySpFaceRequest setParameter:USER_APPID forKey:[IFlyFaceConstant APPID]];
// 以下参数可选，当设置参数 auth_id 时，验证时可通过 auth_id 验证
[self.iFlySpFaceRequest setParameter:@"your_auth_id" forKey:[IFlyFaceConstant AUTH_ID]];
// 以下参数可选，当设置参数property为del时原模型删除后重新注册
[self.iFlySpFaceRequest setParameter:[IFlyFaceConstant DEL] forKey:[IFlyFaceConstant PROPERTY]];
```

回调的定义如下：

```
/**
 * 消息回调
 * @param eventType 消息类型
 * @param params 消息数据对象
 */
- (void) onEvent:(int) eventType WithBundle:(NSString*) params{
    NSLog(@"onEvent | params:%@",params);
}

/**
 * 数据回调，可能调用多次，也可能一次不调用
 * @param buffer 服务端返回的二进制数据
 */
- (void) onData:(NSData*) data{
    NSString *result = [[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding];
    SBJsonParser * jsonParser=[[SBJsonParser alloc] init];
    NSDictionary* dic=[jsonParser objectForKey:result];

    if(dic){
        NSString* strSessionType=[dic objectForKey:@"sst"];
        //注册
    }
}
```

```
        if([strSessionType isEqualToString:@"reg"]){
            [self praseRegResult:result];
        }
        //验证
        if([strSessionType isEqualToString:@"verify"]){
            [self praseVerifyResult:result];
        }
        //检测
        if([strSessionType isEqualToString:@"detect"]){
            [self praseDetectResult:result];
        }

        //关键点
        if([strSessionType isEqualToString:@"align"]){
            [self praseAlignResult:result];
        }
    }
    [result release];
    [jsonParser release];
}

/**
 * 结束回调，没有错误时，error为null
 * @param error 错误类型
 */
- (void) onCompleted:(IFlyFaceError*) error{
}
```

4.1.2 人脸验证

```
// 设置验证参数
[self.iFlySpFaceRequest setParameter:[IFlyFaceConstant VERIFY] forKey:[IFlyFaceConstant SST]];
[self.iFlySpFaceRequest setParameter:USER_APPID forKey:[IFlyFaceConstant APPID]];
//auth_id 用于第三方跟用户关联，由第三方传入和管理
[self.iFlySpFaceRequest setParameter:USER_APPID forKey:[IFlyFaceConstant AUTH_ID]];
//若第三方未传入auth_id 则需要传入gid 来验证，详情参考Demo示例
NSUserDefaults* userDefaults=[NSUserDefaults standardUserDefaults];
NSString* gid=[userDefaults objectForKey:@"gid"];
[self.iFlySpFaceRequest setParameter:gid forKey:[IFlyFaceConstant GID]];
[self.iFlySpFaceRequest setParameter:@"2000" forKey:@"wait_time"];
```

注册/验证结果中包含了是否成功、gid 等信息，详细的 JSON 格式请参照附录 [6.1 人脸识别结果说明](#)，具体解析过程详见 IFlyFaceDemo 工程。

4.1.3 人脸检测

```
// 设置人脸检测参数
[self.iFlySpFaceRequest setParameter:[IFlyFaceConstant DETECT] forKey:[IFlyFaceConstant SST]];
[self.iFlySpFaceRequest setParameter:USER_APPID forKey:[IFlyFaceConstant APPID]];
```

4.1.4 人脸聚焦

```
// 设置人脸聚焦参数
[self.iFlySpFaceRequest setParameter:[IFlyFaceConstant ALIGN] forKey:[IFlyFaceConstant SST]];
[self.iFlySpFaceRequest setParameter:USER_APPID forKey:[IFlyFaceConstant APPID]];
```

4.2 详细 demo

详细的代码可以参考 Demo 中的 `ViewController` 的示例代码。

4.3 使用注意

人脸识别的使用，需要设置的参数和准备工作相对比较简单，但在使用之前建议仔细阅读下 demo 的代码，尤其需要注意服务启动的流程。

5 Q&A

参见论坛：[语音云开放平台](#)。

6 附录

6.1 人脸识别结果说明

json 字段	类型	说明
sst	string	指定本路会话是属于何种性质

gid	string	图像模型 id
rst	bool	结果
sid	string	会话 id
ret	int	错误码
uid	string	用户 id
score	float	打分
pose	dictionary	面部朝向
confidence	float	置信度
position	dictionary	面部的矩形区域
landmark	dictionary	关键点数组

注册时响应结果返回 json 格式串:

```
{
  "sst": "reg",
  "gid": "4a6c124ed6b78436ee5aac4563f13eb5",
  "rst": "success",
  "sid": "wfr2717002e@hf4aed073fdccb4af700",
  "ret": "0",
  "uid": "123456"
}
```

验证时响应结果返回 json 格式串:

```
{
  "gid": "wfr278722e9@ch51370817c1f0477300",
  "ret": "0",
  "rst": "success",
  "score": "83.470",
  "sid": "wfr279e22e8@ch51370817c1fb477400",
  "sst": "verify",
  "uid": "a1644276827",
  "verf": true
}
```

人脸检测时响应结果返回 json 格式串:

```
{
  "face": [
    {
      "attribute": {
        "pose": {
```

```
        "pitch": 1
      },
    },
    "confidence": "10.412",
    "position": {
      "bottom": 447,
      "left": 140,
      "right": 419,
      "top": 168
    },
    "tag": ""
  }
],
"ret": "0",
"rst": "success",
"sid": "wfr278422e9@ch47fc0817c22e477000",
"sst": "detect",
"uid": "a1644276827"
}
```

关键点检测时响应结果返回 json 格式串：

```
{
  "result": [
    {
      "landmark": {
        "left_eye_center": {
          "x": "209.739",
          "y": "229.428"
        },
        "left_eye_left_corner": {
          "x": "177.219",
          "y": "230.914"
        },
        "left_eye_right_corner": {
          "x": "235.839",
          "y": "236.793"
        },
        "left_eyebrow_left_corner": {
          "x": "155.253",
          "y": "187.392"
        },
        "left_eyebrow_middle": {
          "x": "199.240",

```



```
        "y": "182.701"
    },
    "left_eyebrow_right_corner": {
        "x": "246.582",
        "y": "192.358"
    },
    "mouth_left_corner": {
        "x": "204.203",
        "y": "386.777"
    },
    "mouth_lower_lip_bottom": {
        "x": "262.768",
        "y": "416.832"
    },
    "mouth_middle": {
        "x": "263.705",
        "y": "390.507"
    },
    "mouth_right_corner": {
        "x": "317.841",
        "y": "390.864"
    },
    "mouth_upper_lip_top": {
        "x": "264.736",
        "y": "367.996"
    },
    "nose_bottom": {
        "x": "267.811",
        "y": "339.358"
    },
    "nose_left": {
        "x": "225.449",
        "y": "319.586"
    },
    "nose_right": {
        "x": "308.086",
        "y": "323.936"
    },
    "nose_top": {
        "x": "271.755",
        "y": "310.934"
    },
    "right_eye_center": {
        "x": "335.608",
```

```
        "y": "234.335"
      },
      "right_eye_left_corner": {
        "x": "306.995",
        "y": "238.703"
      },
      "right_eye_right_corner": {
        "x": "364.231",
        "y": "240.307"
      },
      "right_eyebrow_left_corner": {
        "x": "300.652",
        "y": "194.243"
      },
      "right_eyebrow_middle": {
        "x": "347.711",
        "y": "188.787"
      },
      "right_eyebrow_right_corner": {
        "x": "391.572",
        "y": "197.455"
      }
    }
  },
  "ret": "0",
  "rst": "success",
  "sid": "wfr278522e9@ch47fc0817c255477600",
  "sst": "align",
  "uid": "a1644276827"
}
```

6.2 错误码列表

- 1、10000~19999 的错误码参见 [MSC 错误码链接](#)。
- 2、其它错误码可通过 IFlyFaceError 获得相应代表错误含义。