
中国移动通信集团公司

CMCC

中国移动 MSC 新手指南（iOS）

目录

1	概述	I
2	MSC 概述	I
2.1	iOS 语音控件 SDK 接口和体系结构	I
2.2	iOS 语音控件 SDK 资源占用描述	II
3	搭建开发环境	II
3.1	创建 iOS 工程	II
3.2	添加静态库	II
3.3	添加 FRAMEWORK	IV
3.4	确认 SDK 的路径	IV
3.5	导入头文件	V
3.6	集成帮助文档到 XCODE	VI
4	在代码中使用开发工具包	VII
4.1	初始化	VII
4.2	语音转写（带界面示例）	VIII
4.3	语音合成（不带界面示例）	9
4.4	语义示例	10
4.5	上传联系人	10
4.6	用户词表的上传	11
4.7	ABNF 语法上传示例	12
4.8	声纹识别	12
5	Q&A	16
6	附录	16
6.1	语义识别结果说明	16
6.2	个性发音人列表	17
6.3	错误码列表	18

1 概述

本文档是开发中国移动 iOS 语音控件 SDK 的用户指南，定义了语音听写、语音合成、语义理解以及个性化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语音 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语音合成、语音听写和语法识别服务。

2 MSC 概述

2.1 iOS 语音控件 SDK 接口和体系结构

本文档是开发中国移动 iOS 语音程序的用户指南，定义了语音听写、语音识别、语音合成以及语义理解相关接口的使用说明和体系结构，如图 3-1 所示。

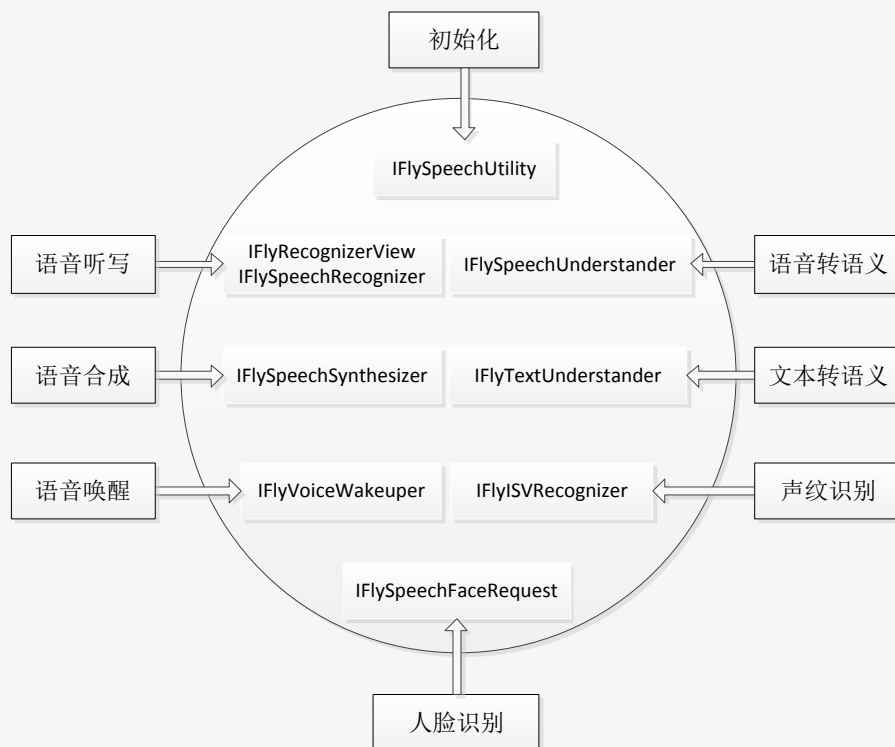


图 3-1 MSC 功能结构图

2.2 iOS 语音控件 SDK 资源占用描述

静态库大小	有 UI	内存占用	有 UI
编译前	11.7M	识别	17.4M
编译后	2.6M	合成	12.8M

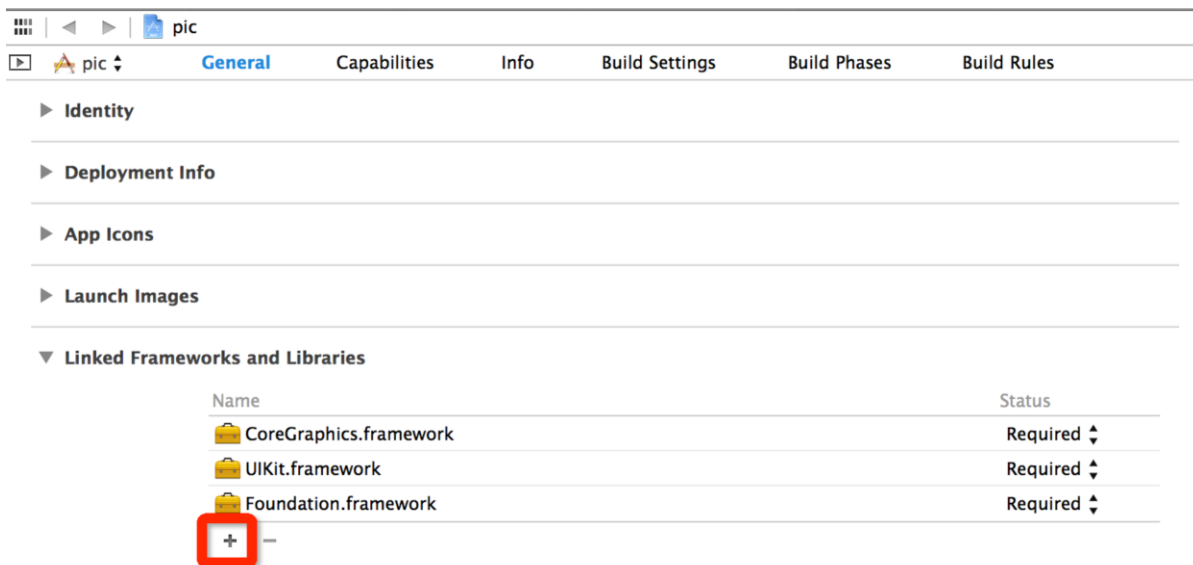
3 搭建开发环境

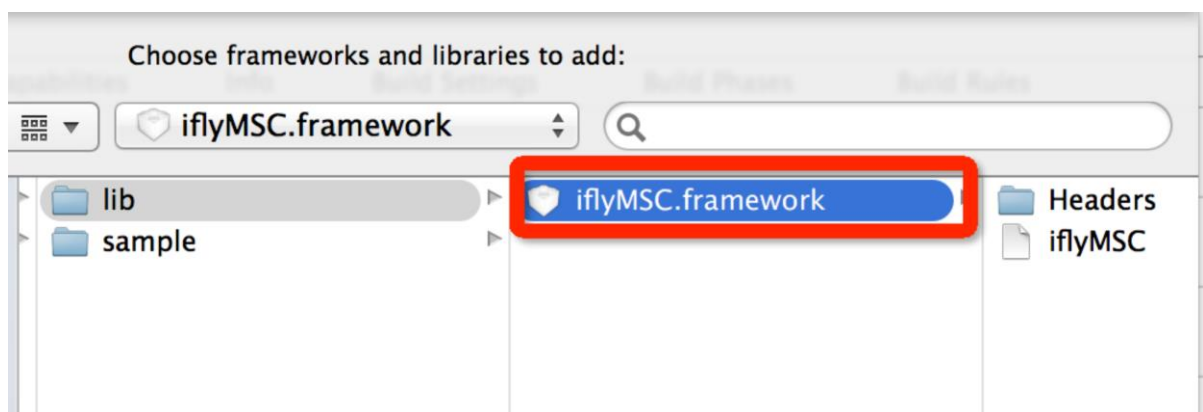
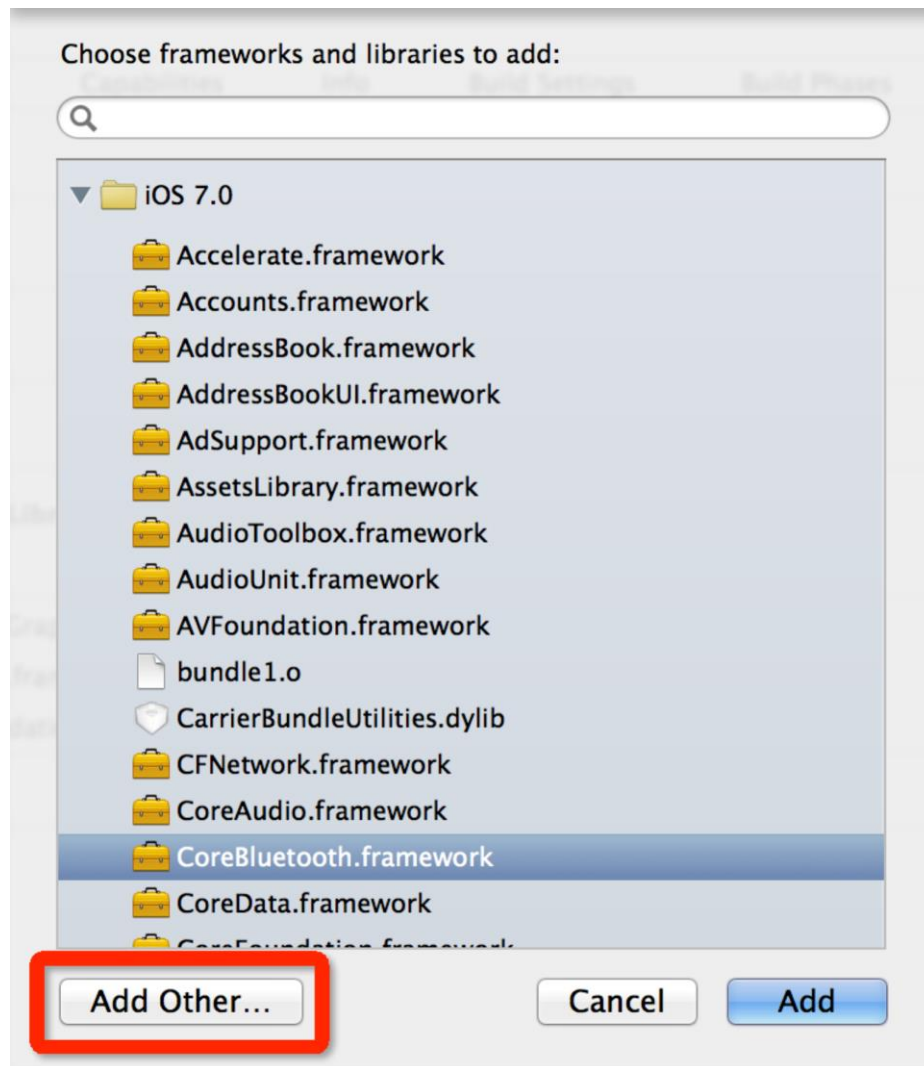
3.1 创建 iOS 工程

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 iflyMSC.framework 添加到新建工程中（如下图所示）。















▼ Linked Frameworks and Libraries

Name	Status
 iflyMSC.framework	Required ▾
 CoreGraphics.framework	Required ▾
 UIKit.framework	Required ▾
 Foundation.framework	Required ▾
+ -	

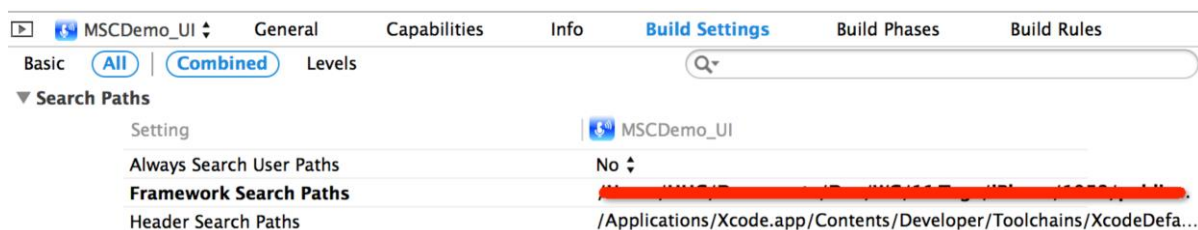
3.3 添加 framework

按下图添加 SDK 所需要的 iOS 库

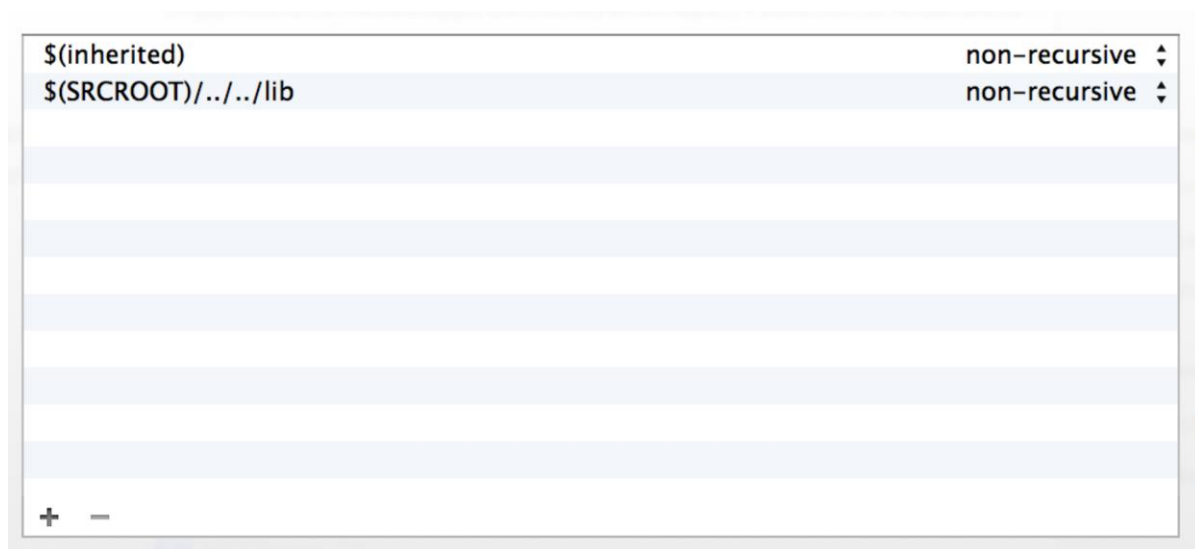
▼ Linked Frameworks and Libraries

Name	Status
 iflyMSC.framework	Required ▾
 AVFoundation.framework	Required ▾
 AddressBook.framework	Required ▾
 SystemConfiguration.framework	Required ▾
 AudioToolbox.framework	Required ▾
 QuartzCore.framework	Required ▾
 libz.dylib	Required ▾
 UIKit.framework	Required ▾
 Foundation.framework	Required ▾
 CoreGraphics.framework	Required ▾

3.4 确认 SDK 的路径



请确认上图红色部分的路径能够找到 iflyMSC.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 MSC 服务的文件中导入相应的头文件

例如：

//带界面的语音识别控件

```
#import "iflyMSC/IFlyRecognizerViewDelegate.h"
```

```
#import "iflyMSC/IFlyRecognizerView.h"
```

//不带界面的语音识别控件

```
#import "iflyMSC/IFlySpeechRecognizerDelegate.h"
```

```
#import "iflyMSC/IFlySpeechRecognizer.h"
```

//不带界面的语音合成控件

```
#import "iflyMSC/IFlySpeechSynthesizerDelegate.h"
```

```
#import "iflyMSC/IFlySpeechSynthesizer.h"
```

3.6 集成帮助文档到 Xcode

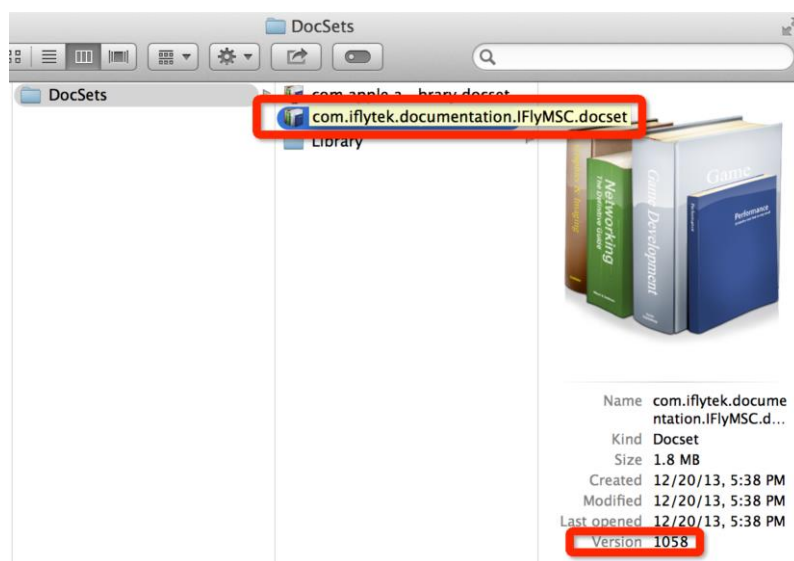
打开终端（terminal 或 iterm），cd 到压缩包的 doc 目录，执行以下命令：

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

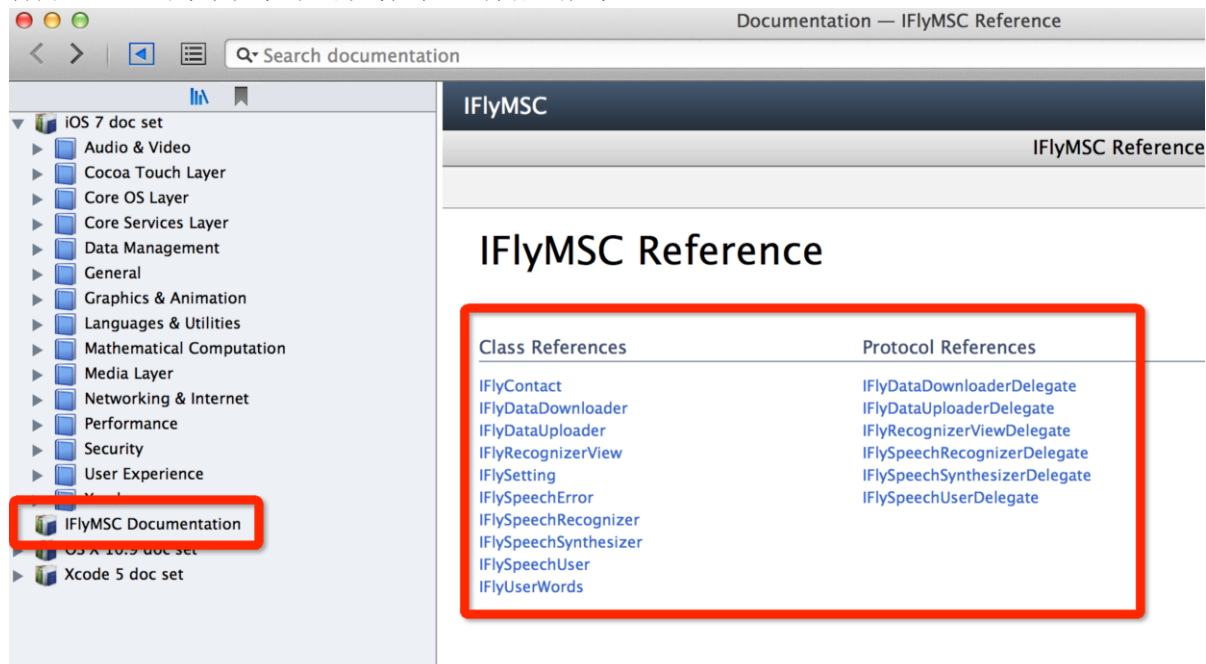
然后执行命令

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

请核对文档的版本为最新下载的版本



打开 Xcode 的帮助文档就可以看到已经集成的文档



4 在代码中使用开发工具包

4.1 初始化

创建用户语音配置对象后才可以使用语音服务，建议在程序入口处调用。

```
//将“12345678”替换成您申请的APPID，申请地址：http://dev.10086.cn/  
  
NSString *initString = [[NSString alloc] initWithFormat:@"appid=%@", @" 12345678"];  
  
[IFlySpeechUtility createUtility:initString];
```

4.2 语音转写（带界面示例）

使用示例如下图所示。

```
//头文件定义
//需要实现IFlyRecognizerViewDelegate，为识别会话的服务代理

@interface RecognizerViewController : UIViewController<IFlyRecognizerViewDelegate>
{
    IFlyRecognizerView      *_iflyRecognizerView;
}

//初始化语音识别控件
_iflyRecognizerView = [[IFlyRecognizerView alloc] initWithCenter:self.view.center];
_iflyRecognizerView.delegate = self;
[_iflyRecognizerView setParameter:@"iat" forKey:[IFlySpeechConstant IFLY_DOMAIN]];
//asr_audio_path保存录音文件名，如不再需要，设置value为nil表示取消，默认目录是documents
[_iflyRecognizerView setParameter:@"asrview.pcm" forKey:[IFlySpeechConstant
ASR_AUDIO_PATH]];

//启动识别服务
[_iflyRecognizerView start];

/*识别结果返回代理
@param resultArray 识别结果
@ param isLast 表示是否最后一次结果
*/
- (void)onResult: (NSArray *)resultArray isLast:(BOOL) isLast
{
}

/*识别会话错误返回代理
@ param  error 错误码
*/
- (void)onError: (IFlySpeechError *) error
{
}
```

4.3 语音合成（不带界面示例）

使用示例如下图所示。

```
//头文件定义
//需要实现IFlySpeechSynthesizerDelegate，为合成会话的服务代理
@interface TTSTableViewController : UIViewController<IFlySpeechSynthesizerDelegate>
{
    IFlySpeechSynthesizer * _iFlySpeechSynthesizer;
}
// 创建合成对象，为单例模式
_iFlySpeechSynthesizer = [IFlySpeechSynthesizer sharedInstance];
_iFlySpeechSynthesizer.delegate = self;

//设置语音合成的参数
//语速,取值范围 0~100
[_iFlySpeechSynthesizer setParameter:@"50" forKey:[IFlySpeechConstant SPEED]];
//音量;取值范围 0~100
[_iFlySpeechSynthesizer setParameter:@"50" forKey:[IFlySpeechConstant VOLUME]];
//发音人,默认为"xiaoyan";可以设置的参数列表可参考个性化发音人列表
[_iFlySpeechSynthesizer setParameter:@"xiaoyan" forKey:[IFlySpeechConstant VOICE_NAME]];
//音频采样率,目前支持的采样率有 16000 和 8000
[_iFlySpeechSynthesizer setParameter:@"8000" forKey:[IFlySpeechConstant SAMPLE_RATE]];
//asr_audio_path保存录音文件路径，如不再需要，设置value为nil表示取消，默认目录是 documents
[_iFlySpeechSynthesizer setParameter:@"tts.pcm" forKey:[IFlySpeechConstant TTS_AUDIO_PATH]];

//启动合成会话

[_iFlySpeechSynthesizer startSpeaking:@"你好，我是中国移动的小燕"];

//合成结束，此代理必须要实现
- (void) onCompleted:(IFlySpeechError *) error{}
//可选接口
- (void) onSpeakBegin{
//合成开始}
- (void) onBufferProgress:(int) progress message:(NSString *)msg{
//合成缓冲进度}
- (void) onSpeakProgress:(int) progress{
//合成播放进度}
```

4.4 语义示例

使用示例如下图所示。

```
//注：如需使用语义功能，请先将需求直接发送邮件到灵犀云客服邮箱lingxicloud@139.com进行审核配置
_iFlySpeechUnderstander = [IFlySpeechUnderstander sharedInstance];
_iFlySpeechUnderstander.delegate = self;

[_iFlySpeechUnderstander startListening]; //启动识别服务

/*语义识别结果返回代理
@param resultArray 识别结果
@ param isLast 表示是否最后一次结果
*/
- (void) onResults:(NSArray *) results isLast:(BOOL) isLast
{
}
- (void) onError:(IFlySpeechError*) error
{
}
-
```

4.5 上传联系人

使用示例如下图所示。

```
//创建上传对象
_uploader = [[IFlyDataUploader alloc] init];

//获取联系人集合
IFlyContact *iFlyContact = [[IFlyContact alloc] init];
NSString *contactList = [iFlyContact contact];

//设置参数
[_uploader setParameter:@"uup" forKey:@"subject"];
[_uploader setParameter:@"contact" forKey:@"dt"];

//启动上传
[_uploader uploadDataWithCompletionHandler:^(NSString * grammarID, IFlySpeechError *error)
{
    //接受返回的grammarID和error
    [self onUploadFinished:grammarID error:error];

}name:@"contact" data: contactList];
```

4.6 用户词表的上传

使用示例如下图所示。

```
//创建上传对象
_upload = [[IFlyDataUploader alloc] init];

//用户词表
#define USERWORDS    @"{\"userword\": [{\"name\": \"cmcc\", \"words\": [\"德国盐猪手\", \"1912酒吧街\", \"清蒸鲈鱼\", \"挪威三文鱼\", \"黄埔军校\", \"横沙牌坊\", \"中国移动\", \"王小貳\"]}]}"

#define NAME @"userwords"

//生成用户词表对象
IFlyUserWords *iFlyUserWords = [[IFlyUserWords alloc] initWithJson:USERWORDS];

//设置参数
[_uploader setParameter:@"iat" forKey:@"sub"];
[_uploader setParameter:@"userword" forKey:@"dtl"];

[_uploader uploadDataWithCompletionHandler:^(NSString * grammarID, IFlySpeechError *error)
{
    //接受返回的grammarID和error
    [self onUploadFinished:grammarID error:error];
}
```

4.7 abnf 语法上传示例

使用示例如下图所示。

```
// ABNF语法示例，可以说“北京到上海”
#define ABNFPARAM @"sub=asr,dtb=abnf"
#define ABNFDATA = "#ABNF 1.0 gb2312;
                    language zh-CN;
                    mode voice;
                    root $main;
                    $main = $place1 到$place2 ;
                    $place1 = 北京 | 武汉 | 南京 | 天津 | 天京 | 东京;
                    $place2 = 上海 | 合肥;"

//创建上传对象
_uploader = [[IFlyDataUploader alloc] init];

//设置参数
[_uploader setParameter:@"asr" forKey:@"sub"];
[_uploader setParameter:@"abnf" forKey:@"dtb"];

//上传 abnf 语法
[_uploader uploadDataWithCompletionHandler:^(NSString * grammarID, IFlySpeechError
*error)
{
    //接受返回的grammarID和error
    [self setGrammarId:grammarID];

}name:ABNFNAME data:ABNFDATA];
```

4.8 声纹识别

声纹识别，主要是提供基于用户声纹特征的注册、验证服务，语音云平台支持 2 种类型的声纹密码类型，即文本密码和数字密码，在注册时需要指定声纹类型。

//1.创建声纹对象

```
isvRec=[IFlyISVRecognizer sharedInstance];  
isvRec.delegate=self;
```

//2.设置声纹工作参数

```
//设置密码类型，pwdt的取值为1、3，分别表示文本密码和数字密码  
[isvRec setParameter:[NSString stringWithFormat:@"%d",pwdt] forKey:@"pwdt"];
```

pwdt 的取值说明如表 2 所示：

表 2 pwdt 取值与声纹服务类型

pwdt 取值	说明
1	文本密码。用户通过读出指定的文本内容来进行声纹注册和验证，现阶段支持的文本只有“芝麻开门”一种。
3	数字密码。从云端拉取一组特定的数字串（共分 5 组，每组 8 位数字），用户依次读出这 5 组数字进行注册，在验证过程中会生成一串特定的数字，用户通过朗读这串数字进行验证。

密码内容需调用接口从云端获取：

```
//通过调用getPasswordList方法来获取密码。获取密码的时候需指定声纹密码  
类型，pwdt为1表示固定文本密码，pwdt为3表示数字密码，自由说不需从云  
端获取密码。
```

```
// getPasswordList可以参照demo所示。
```

```
NSArray *tmpArray=[isvRec getPasswordList:ivppwdt];
```

获取到密码后，接下来进行声纹注册，即要求用户朗读若干次指定的内容，这一过程也称为声纹模型的训练。

```
// 设置业务类型为训练
[isvRec setParameter:@"train" forKey:@"sst"];
// 设置密码类型
[isvRec setParameter:[NSString stringWithFormat:@"%d",pwdt] forKey:@"pwdt"];
// 对于文本密码和数字密码，必须设置密码的文本内容，ptxt的取值为“我的地盘我做主”、“移动改变生活”、“芝麻开门”或者是从云端拉取的数字密码(每8位用“-”隔开)。自由说略过此步
[isvRec setParameter:ptxt forKey:@"ptxt"];
// 设置声纹对应的auth_id，它用于标识声纹对应的用户
[isvRec setParameter:auth_id forKey:@"auth_id"];
// 设置有效录音时间
[isvRec setParameter:@"3000" forKey:@"vad_timeout"];
// 末端静音检测时间，用于检测到静音自动停止录音
[isvRec setParameter:@"700" forKey:@"vad_speech_tail"];
```

//3.启动训练服务

// 开始注册，当得到注册结果时，SDK会将其封装成NSDictionary对象，回调onResult方法进行处理，处理方法详见Demo示例

```
[isvRec startListening];
```

//4.声纹实现 delegate

```
@protocol IFlyISVDelegate
-(void) onResult:(NSDictionary *)dic;           //训练和验证正常结果返回
-(void) onError:(IFlySpeechError *) errorCode; //会话结束，正常结束错误码为0，非0表示有错误，错误码见下面描述
@optional
-(void) onRecognition;                          //结果处理中
-(void) onVolumeChanged: (int)volume;           //录音音量发生改变
@end
```

推荐在注册声纹模型时每个用户都指定一个唯一的 auth_id。auth_id 的格式为:6-18 个字符，为字母、数字和下划线的组合且必须以字母开头，不支持中文字符，不能包含空格。

开发者通过重写 onResult 方法来处理注册和验证结果。在结果 result 中携带错误码，用来判别注册是否成功以及出错原因，部分错误码的含义如表 3 所示：

表 3 onResult 返回的部分错误码说明

错误码	错误值	说明
-----	-----	----

MSS_ERROR_IVP_GENERAL	11600	正常，请继续传音频
MSS_ERROR_IVP_EXTRA_RGN_SOPPORT	11601	rgn 超过最大支持次数 9
MSS_ERROR_IVP_TRUNCATED	11602	音频波形幅度太大，超出系统范围，发生截幅
MSS_ERROR_IVP_MUCH_NOISE	11603	太多噪音
MSS_ERROR_IVP_TOO_LOW	11604	声音太小
MSS_ERROR_IVP_ZERO_AUDIO	11605	没检测到音频
MSS_ERROR_IVP_UTTER_TOO_SHORT	11606	音频太短
MSS_ERROR_IVP_TEXT_NOT_MATCH	11607	音频内容与给定文本不一致
MSS_ERROR_IVP_NO_ENOUGH_AUDIO	11608	音频长达不到自由说的要求

1) 声纹验证

声纹验证过程与声纹注册类似，不同之处仅在于@“sst”参数需要设置为@“verify”，其他参数的设置、验证结果的处理过程可参考上一节。

另外，为了达到较好的效果，请在声纹注册与验证过程中尽量与麦克风保持同样的距离（建议的最佳距离是 15 厘米左右）。若距离较远，可能会对验证通过率产生较大影响。

2) 声纹模型操作

声纹注册成功后，在云端会生成一个对应的模型来存储声纹信息，声纹模型的操作即对模型进行查询和删除。

```
// 开发者调用sendRequest方法查询或者删除模型，该函数的定义如下：
-(BOOL) sendRequest:
    (NSString*)cmd
    authid:(NSString *)auth_id
    pwdt:(int)pwdt
    ptxt:(NSString *)ptxt
    vid:(NSString *)vid
    err:(int *)err;

//cmd: @"query"表示查询，@"del"表示删除
//auth_id表示用户名;
//pwdt表示声纹类型;
//ptxt表示查询或者删除的密码文本;
//vid是用户注册成功后服务器返回的32位标识，查询和删除时，vid可以设置位nil;
//err是查询的错误码。 通常查询或者删除成功，该函数会返回YES,否则返回
```

3) 声纹业务返回结果格式和参数说明

文本密码 JSON 示例

```
{"txt_pwd":["芝麻开门"]}
```

数字密码 JSON 示例

```
{"num_pwd":["03285469","09734658","53894276","57392804","68294073"]}
```

声纹业务结果（NSDictionary 类型结果）成员说明

成员名	参数解释
sst	业务类型，取值为 train 或 verify
ret	返回值，0 为成功，-1 为失败
vid	注册成功的声纹模型 id
score	当前声纹相似度
suc	本次注册已成功的训练次数
rgn	本次注册需要的训练次数
trs	注册完成描述信息
err	注册/验证返回的错误码
dcs	描述信息

5 Q&A

参见开发者网站：http://dev.10086.cn/wiki/?p4_02_04。

6 附录

6.1 语义识别结果说明

json 字段	英文全称	类型	说明
sn	sentence	int	第几句
ls	last sentence	boolean	是否最后一句
bg	begin	int	开始
ed	end	int	结束
ws	words	array	词

cw	chinese word	array	中文分词
w	word	string	单字
sc	socre	int	分数

转写结果示例：

```
{
  "sn":1,"ls":true,"bg":0,"ed":0,"ws":[{
    "bg":0,"cw":{"w":"今天","sc":0}},
    {"bg":0,"cw":{"w":"的","sc":0}},
    {"bg":0,"cw":{"w":"天气","sc":0}},
    {"bg":0,"cw":{"w":"怎么","sc":0}},
    {"bg":0,"cw":{"w":"样","sc":0}},
    {"bg":0,"cw":{"w":"。","sc":0}}
  ]
}
```

多候选结果示例：

```
{
  "sn":1,"ls":false,"bg":0,"ed":0,"ws":[
    {"bg":0,"cw":{"w":"我想听","sc":0}},
    {"bg":0,"cw":{"w":"拉德斯基进行曲","sc":0}},
    {"bg":0,"cw":{"w":"拉得斯进行曲","sc":0}}
  ]
}
```

语法识别结果示例：

```
{
  "sn":1,"ls":true,"bg":0,"ed":0,"ws":[
    {
      "bg":0,"cw":{
        {"sc":"70","gm":"0","w":"北京到上海"},
        {"sc":"69","gm":"0","w":"天京到上海"},
        {"sc":"58","gm":"0","w":"东京到上海"}
      }
    }
  ]
}
```

6.2 个性发音人列表

- 1、语言为中英文的发音人可以支持中英文的混合朗读。
- 2、英文发音人只能朗读英文，中文无法朗读。
- 3、汉语发音人只能朗读中文，遇到英文会以单个字母的方式进行朗读。

发音人名称	属性	语言	参数名称	备注
小燕	青年女声	中英文（普通话）	xiaoyan	默认
小宇	青年男声	中英文（普通话）	xiaoyu	
凯瑟琳	青年女声	英文	Catherine	
亨利	青年男声	英文	henry	
玛丽	青年女声	英文	vimary	
小研	青年女声	中英文（普通话）	vixy	

小琪	青年女声	中英文（普通话）	vixq	
小峰	青年男声	中英文（普通话）	vixf	
小梅	青年女声	中英文（粤语）	vixm	
小莉	青年女声	中英文（台湾普通话）	vixl	
小蓉	青年女声	汉语（四川话）	vixr	
小芸	青年女声	汉语（东北话）	vixyun	
小坤	青年男声	汉语（河南话）	vixk	
小强	青年男声	汉语（湖南话）	vixqa	
小莹	青年女声	汉语（陕西话）	vixying	
小新	童年男声	汉语（普通话）	vixx	
楠楠	童年女声	汉语（普通话）	vinn	
老孙	老年男声	汉语（普通话）	vils	

6.3 错误码列表

1、10000~19999 的错误码参见如下。

错误码	错误值	意义
MSP_SUCCESS	0	函数执行成功
MSP_ERROR_FAIL	-1	失败
MSP_ERROR_EXCEPTION	-2	异常
MSP_ERROR_GENERAL	10100	基码
MSP_ERROR_OUT_OF_MEMORY	10101	内存越界
MSP_ERROR_FILE_NOT_FOUND	10102	文件没有发现
MSP_ERROR_NOT_SUPPORT	10103	不支持
MSP_ERROR_NOT_IMPLEMENT	10104	没有实现
MSP_ERROR_ACCESS	10105	没有权限
MSP_ERROR_INVALID_PARA	10106	无效的参数
MSP_ERROR_INVALID_PARA_VALUE	10107	无效的参数值
MSP_ERROR_INVALID_HANDLE	10108	无效的句柄
MSP_ERROR_INVALID_DATA	10109	无效的数据
MSP_ERROR_NO_LICENSE	10110	没有授权许可
MSP_ERROR_NOT_INIT	10111	没有初始化
MSP_ERROR_NULL_HANDLE	10112	空句柄
MSP_ERROR_OVERFLOW	10113	溢出
MSP_ERROR_TIME_OUT	10114	超时
MSP_ERROR_OPEN_FILE	10115	打开文件出错
MSP_ERROR_NOT_FOUND	10116	没有发现
MSP_ERROR_NO_ENOUGH_BUFFER	10117	没有足够的内存
MSP_ERROR_NO_DATA	10118	没有数据
MSP_ERROR_NO_MORE_DATA	10119	没有更多的数据

错误码	错误值	意义
MSP_ERROR_SKIPPED	10120	跳过
MSP_ERROR_ALREADY_EXIST	10121	已经存在
MSP_ERROR_LOAD_MODULE	10122	加载模块失败
MSP_ERROR_BUSY	10123	忙碌
MSP_ERROR_INVALID_CONFIG	10124	无效的配置项
MSP_ERROR_VERSION_CHECK	10125	版本错误
MSP_ERROR_CANCELED	10126	取消
MSP_ERROR_INVALID_MEDIA_TYPE	10127	无效的媒体类型
MSP_ERROR_CONFIG_INITIALIZE	10128	初始化 Config 实例
MSP_ERROR_CREATE_HANDLE	10129	建立句柄
MSP_ERROR_CODING_LIB_NOT_LOAD	10130	编解码库未加载
MSP_ERROR_NET_GENERAL	10200	网络一般错误
MSP_ERROR_NET_OPENSOCK	10201	打开套接字
MSP_ERROR_NET_CONNECTSOCK	10202	套接字连接
MSP_ERROR_NET_ACCEPTSOCK	10203	套接字接收
MSP_ERROR_NET_SENDSOCK	10204	发送
MSP_ERROR_NET_RECVSOCK	10205	接收
MSP_ERROR_NET_INVALIDSOCK	10206	无效的套接字
MSP_ERROR_NET_BADADDRESS	10207	无效的地址
MSP_ERROR_NET_BINDSEQUENCE	10208	绑定次序
MSP_ERROR_NET_NOTOPENSOCK	10209	套接字没有打开
MSP_ERROR_NET_NOTBIND	10210	没有绑定
MSP_ERROR_NET_NOTLISTEN	10211	没有监听
MSP_ERROR_NET_CONNECTCLOSE	10212	连接关闭
MSP_ERROR_NET_NOTDGRAMSOCK	10213	非数据报套接字
MSP_ERROR_NET_DNS	10214	DNS 解析错误
MSP_ERROR_MSG_GENERAL	10300	消息一般错误
MSP_ERROR_MSG_PARSE_ERROR	10301	解析
MSP_ERROR_MSG_BUILD_ERROR	10302	构建
MSP_ERROR_MSG_PARAM_ERROR	10303	参数出错
MSP_ERROR_MSG_CONTENT_EMPTY	10304	Content 为空
MSP_ERROR_MSG_INVALID_CONTENT_TYPE	10305	Content 类型无效
MSP_ERROR_MSG_INVALID_CONTENT_LENGTH	10306	Content 长度无效
MSP_ERROR_MSG_INVALID_CONTENT_ENCODE	10307	Content 编码无效
MSP_ERROR_MSG_INVALID_KEY	10308	Key 无效
MSP_ERROR_MSG_KEY_EMPTY	10309	Key 为空
MSP_ERROR_MSG_SESSION_ID_EMPTY	10310	会话 ID 为空
MSP_ERROR_MSG_LOGIN_ID_EMPTY	10311	登录 ID 为空
MSP_ERROR_MSG_SYNC_ID_EMPTY	10312	同步 ID 为空
MSP_ERROR_MSG_APP_ID_EMPTY	10313	应用 ID 为空

错误码	错误值	意义
MSP_ERROR_MSG_EXTERN_ID_EMPTY	10314	扩展 ID 为空
MSP_ERROR_MSG_INVALID_CMD	10315	无效的命令
MSP_ERROR_MSG_INVALID_SUBJECT	10316	无效的主题
MSP_ERROR_MSG_INVALID_VERSION	10317	无效的版本
MSP_ERROR_MSG_NO_CMD	10318	没有命令
MSP_ERROR_MSG_NO_SUBJECT	10319	没有主题
MSP_ERROR_MSG_NO_VERSION	10320	没有版本号
MSP_ERROR_MSG_MSSP_EMPTY	10321	消息为空
MSP_ERROR_MSG_NEW_RESPONSE	10322	新建响应消息失败
MSP_ERROR_MSG_NEW_CONTENT	10323	新建 Content 失败
MSP_ERROR_MSG_INVALID_SESSION_ID	10324	无效的会话 ID
MSP_ERROR_DB_GENERAL	10400	数据库一般错误
MSP_ERROR_DB_EXCEPTION	10401	异常
MSP_ERROR_DB_NO_RESULT	10402	没有结果
MSP_ERROR_DB_INVALID_USER	10403	无效的用户
MSP_ERROR_DB_INVALID_PWD	10404	无效的密码
MSP_ERROR_DB_CONNECT	10405	连接出错
MSP_ERROR_DB_INVALID_SQL	10406	无效的 SQL
MSP_ERROR_RES_GENERAL	10500	资源一般错误
MSP_ERROR_RES_LOAD	10501	没有加载
MSP_ERROR_RES_FREE	10502	空闲
MSP_ERROR_RES_MISSING	10503	缺失
MSP_ERROR_RES_INVALID_NAME	10504	无效的名称
MSP_ERROR_RES_INVALID_ID	10505	无效的 ID
MSP_ERROR_RES_INVALID_IMG	10506	无效的映像
MSP_ERROR_RES_WRITE	10507	写操作
MSP_ERROR_RES_LEAK	10508	泄露
MSP_ERROR_RES_HEAD	10509	资源头部错误
MSP_ERROR_RES_DATA	10510	数据出错
MSP_ERROR_RES_SKIP	10511	跳过
MSP_ERROR_TTS_GENERAL	10600	合成一般错误
MSP_ERROR_TTS_TEXTEND	10601	文本结束
MSP_ERROR_TTS_TEXT_EMPTY	10602	文本为空
MSP_ERROR_REC_GENERAL	10700	一般错误
MSP_ERROR_REC_INACTIVE	10701	处于不活跃状态
MSP_ERROR_REC_GRAMMAR_ERROR	10702	语法错误
MSP_ERROR_REC_NO_ACTIVE_GRAMMARS	10703	没有活跃的语法
MSP_ERROR_REC_DUPLICATE_GRAMMAR	10704	语法重复
MSP_ERROR_REC_INVALID_MEDIA_TYPE	10705	无效的媒体类型
MSP_ERROR_REC_INVALID_LANGUAGE	10706	无效的语言

错误码	错误值	意义
MSP_ERROR_REC_URI_NOT_FOUND	10707	没有对应的 URI
MSP_ERROR_REC_URI_TIMEOUT	10708	获取 URI 内容超时
MSP_ERROR_REC_URI_FETCH_ERROR	10709	获取 URI 内容时出错
MSP_ERROR_EP_GENERAL	10800	(EP) 一般错误
MSP_ERROR_EP_NO_SESSION_NAME	10801	(EP) 链接没有名字
MSP_ERROR_EP_INACTIVE	10802	(EP) 不活跃
MSP_ERROR_EP_INITIALIZED	10803	(EP) 初始化出错
MSP_ERROR_LOGIN_SUCCESS	11000	登录成功
MSP_ERROR_LOGIN_NO_LICENSE	11001	无授权
MSP_ERROR_LOGIN_SESSIONID_INVALID	11002	无效的 SessionID
MSP_ERROR_LOGIN_SESSIONID_ERROR	11003	错误的 SessionID
MSP_ERROR_LOGIN_UNLOGIN	11004	未登录
MSP_ERROR_LOGIN_INVALID_USER	11005	无效的用户
MSP_ERROR_LOGIN_INVALID_PWD	11006	无效的密码
MSP_ERROR_LOGIN_SYSTEM_ERROR	11099	系统错误
MSP_ERROR_AUTH_NO_LICENSE	11200	无授权
MSP_ERROR_AUTH_NO_ENOUGH_LICENSE	11201	授权不足
MSP_ERROR_AUTH_INVALID_LICENSE	11202	无效的授权
MSP_ERROR_AUTH_LICENSE_EXPIRED	11203	授权过期
MSP_ERROR_AUTH_NEED_MORE_DATA	11204	无设置信息
MSP_ERROR_AUTH_LICENSE_TO_BE_EXPIRED	11205	授权即将过期警告
MSP_ERROR_AUTH_INVALID_MACHINE_ID	11206	无效的机器码
MSP_ERROR_AUTH_LOCAL_ASR_FORBIDDEN	11207	禁止使用本地设别
MSP_ERROR_AUTH_LOCAL_TTS_FORBIDDEN	11208	禁止使用本地合成
MSP_ERROR_AUTH_LOCAL_IVW_FORBIDDEN	11029	禁止使用本地唤醒
MSP_ERROR_AUTH_ERROR_END	11250	授权相关错误码结束标记
MSP_ERROR_ASE_EXCEP_SILENCE	11401	无语音或音量太小
MSP_ERROR_ASE_EXCEP_SNRATIO	11402	信噪比低或有效语音过短
MSP_ERROR_ASE_EXCEP_PAPERDATA	11403	非试卷数据
MSP_ERROR_ASE_EXCEP_PAPERCONTENTS	11404	试卷内容有误
MSP_ERROR_ASE_EXCEP_NOTMONO	11405	录音格式有误
MSP_ERROR_ASE_EXCEP_OTHERS	11406	其他评测数据异常, 包括错读、漏读、恶意录入、试卷内容等错误
MSP_ERROR_ASE_EXCEP_PAPERFMT	11407	试卷格式有误
MSP_ERROR_ASE_EXCEP_ULISTWORD	11408	存在未登记词
MSS_ERROR_IVP_MODEL_NOT_FOUND_IN_HBASE	11610	声纹模型数据找不到
MSP_ERROR_HTTP_BASE	12000	HTTP 错误基码

2、其它错误码参见下表

错误码	错误值	意义
-----	-----	----

ERROR_NO_NETWORK	20001	无有效的网络连接
ERROR_NETWORK_TIMEOUT	20002	网络连接超时
ERROR_NET_EXPECTATION	20003	网络连接发生异常
ERROR_INVALID_RESULT	20004	无有效的结果
ERROR_NO_MATCH	20005	无匹配结果
ERROR_AUDIO_RECORD	20006	录音失败
ERROR_NO_SPEECH	20007	未检测到语音
ERROR_SPEECH_TIMEOUT	20008	音频输入超时
ERROR_EMPTY_UTTERANCE	20009	无效的文本输入
ERROR_FILE_ACCESS	20010	文件读写失败
ERROR_PLAY_MEDIA	20011	音频播放失败
ERROR_INVALID_PARAM	20012	无效的参数
ERROR_TEXT_OVERFLOW	20013	文本溢出
ERROR_INVALID_DATA	20014	无效数据
ERROR_LOGIN	20015	用户未登陆
ERROR_PERMISSION_DENIED	20016	无效授权
ERROR_INTERRUPT	20017	被异常打断
ERROR_VERSION_LOWER	20018	版本过低
ERROR_COMPONENT_NOT_INSTALLED	21001	没有安装语音组件
ERROR_ENGINE_NOT_SUPPORTED	21002	引擎不支持
ERROR_ENGINE_INIT_FAIL	21003	初始化失败
ERROR_ENGINE_CALL_FAIL	21004	调用失败
ERROR_ENGINE_BUSY	21005	引擎繁忙
ERROR_LOCAL_NO_INIT	22001	本地引擎未初始化
ERROR_LOCAL_RESOURCE	22002	本地引擎无资源
ERROR_LOCAL_ENGINE	22003	本地引擎内部错误
ERROR_IVW_INTERRUPT	22004	本地唤醒引擎被异常打断
ERROR_UNKNOWN	20999	未知错误