
科大讯飞股份有限公司
iFLYTEK CO.,LTD

科大讯飞 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	语音唤醒示例	10
4.5	语义示例	12
4.6	上传联系人	12
4.7	用户词表的上传	13
4.8	ABNF 语法上传示例	14
1	Q&A	14
2	附录	15
2.1	语义识别结果说明	15
2.2	个性发音人列表	16
2.3	错误码列表	16

1 概述

本文档是开发科大讯飞 iOS 语音控件 SDK 的用户指南，定义了语音听写、语音合成、语音唤醒、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

本文档是开发科大讯飞 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、语义理解以及语音唤醒相关接口的使用说明和体系结构，如图 3-1 所示。

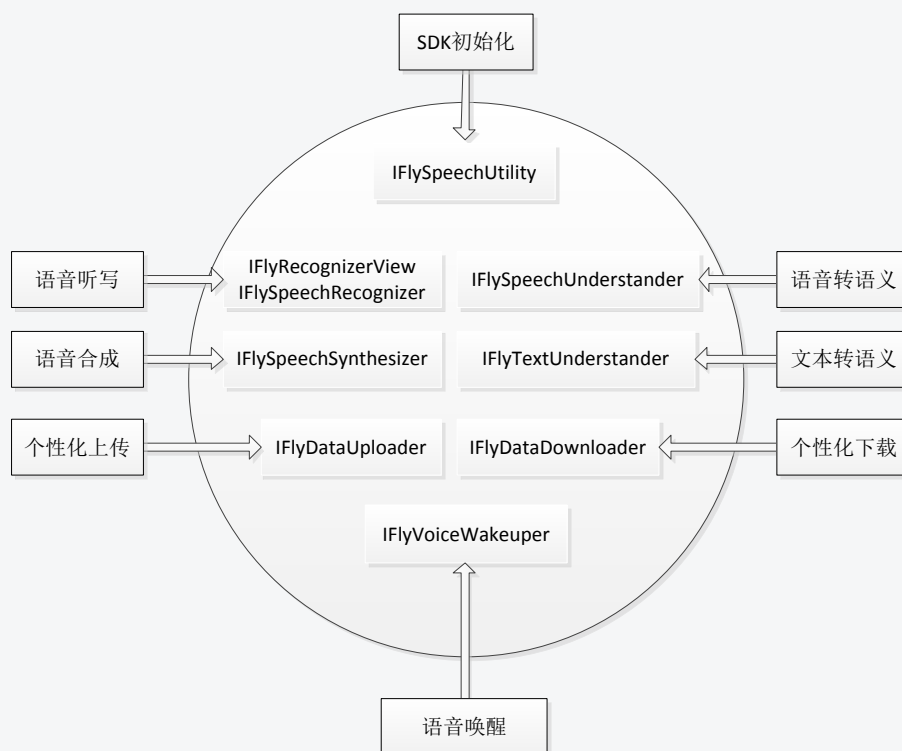


图 3-1 MSC 功能结构图

2.2 iOS 语音控件 SDK 资源占用描述

静态库大小	无 UI	有 UI	内存占用	无 UI	有 UI
编译前	9.9 M	11.7M	识别	14.1M	17.4M
编译后	1.8 M	2.6M	合成	12.3M	12.8M

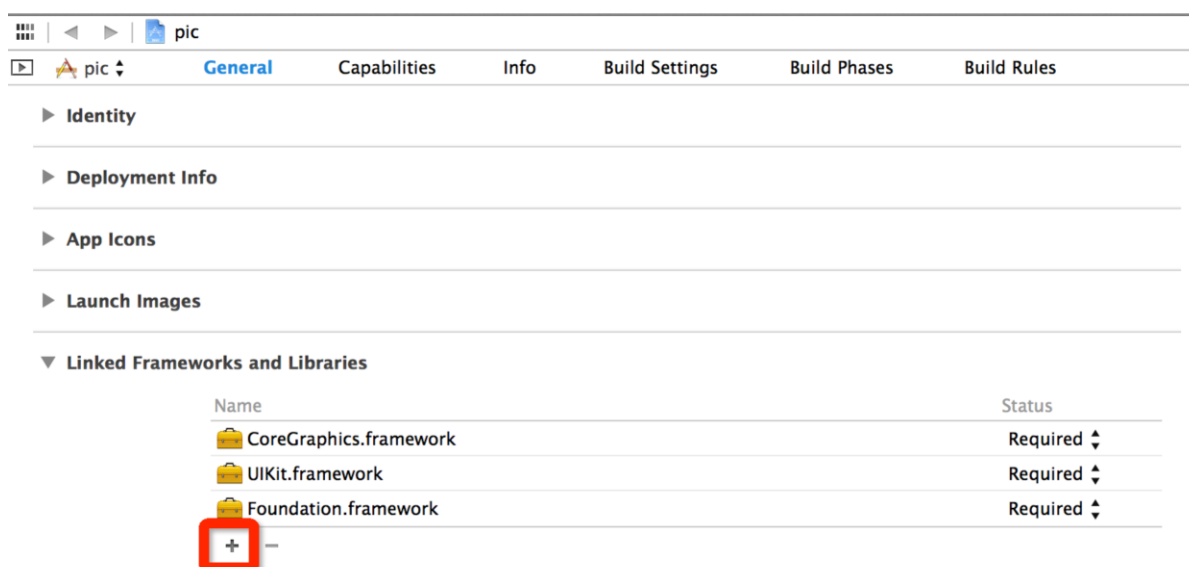
3 搭建开发环境

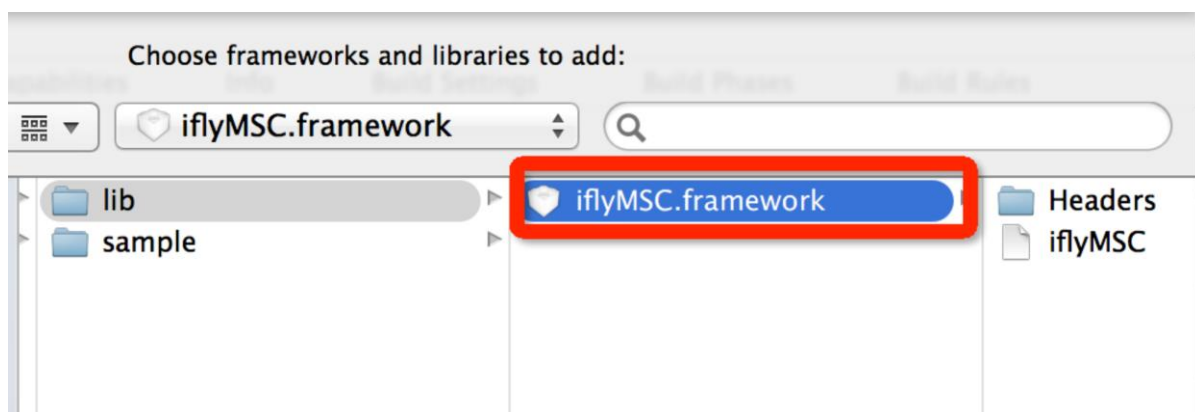
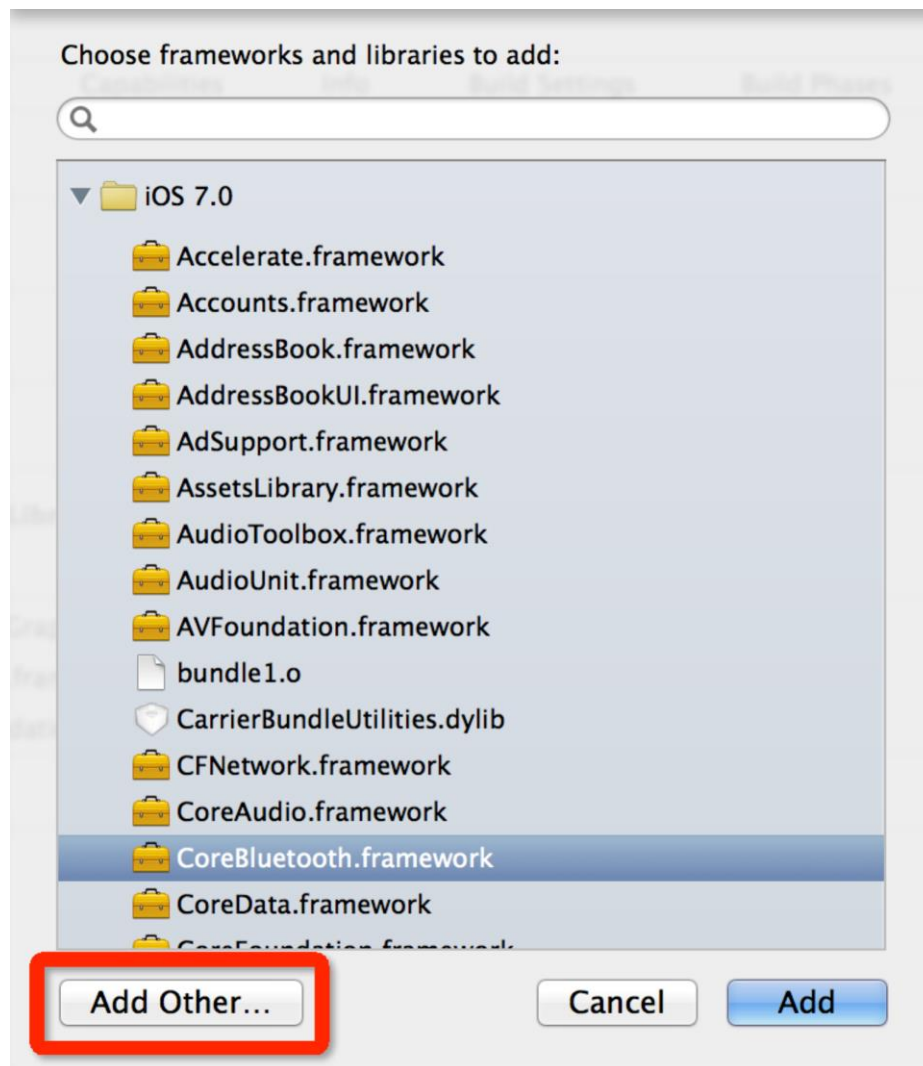
3.1 创建 iOS 工程

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
+ -	

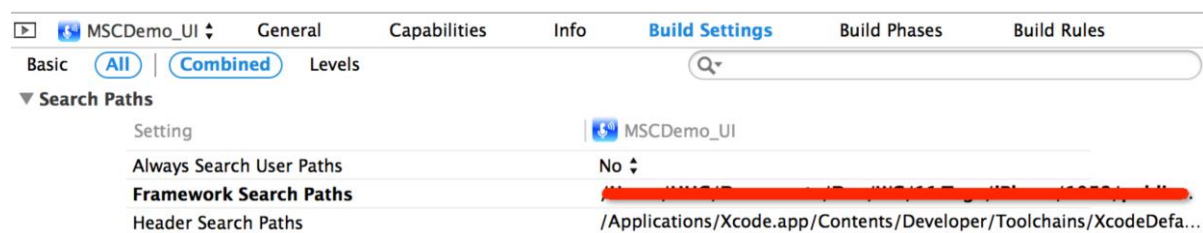
3.3 添加 framework

按下图添加 SDK 所需要的 iOS 库

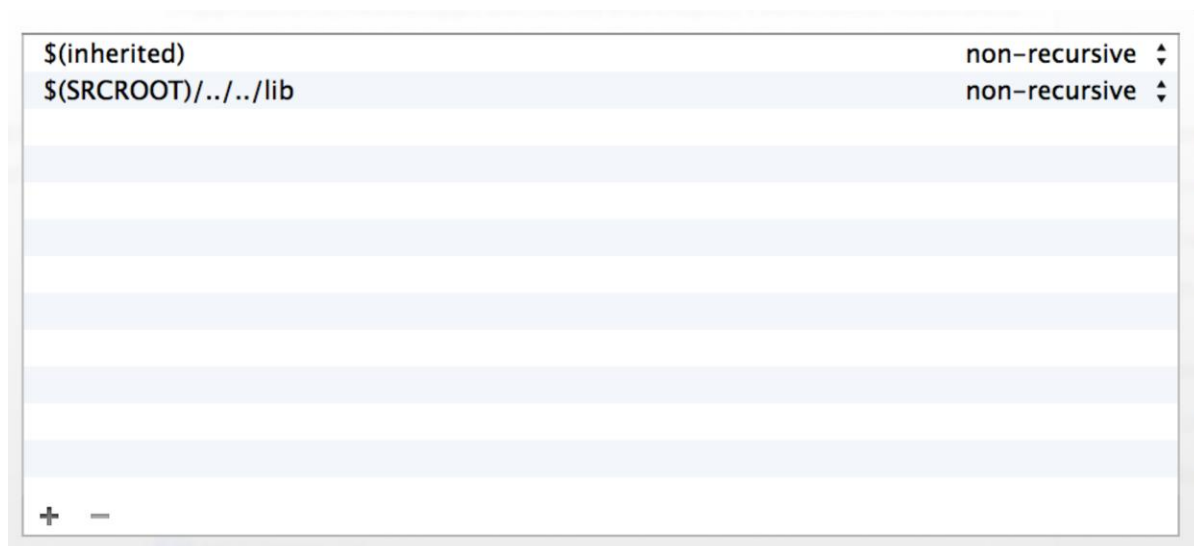
▼ Linked Frameworks and Libraries

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

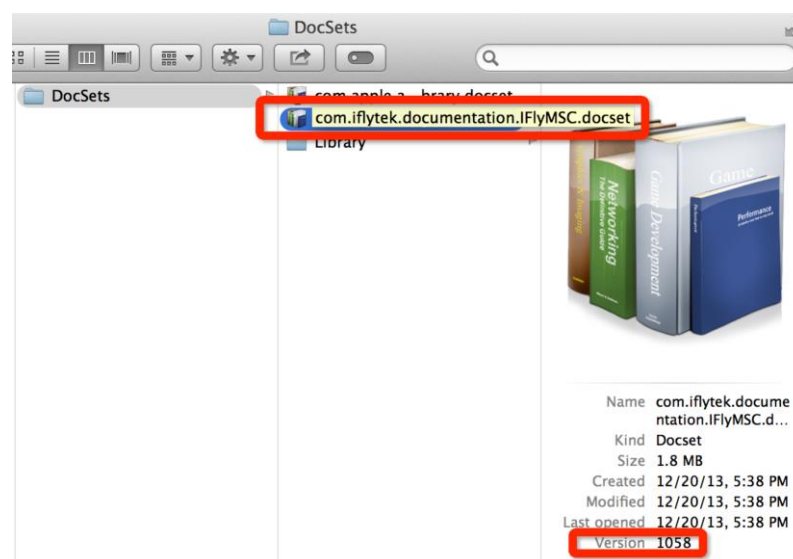
打开终端（terminal 或 iterm），cd 到压缩包的 doc 目录，执行以下命令：

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

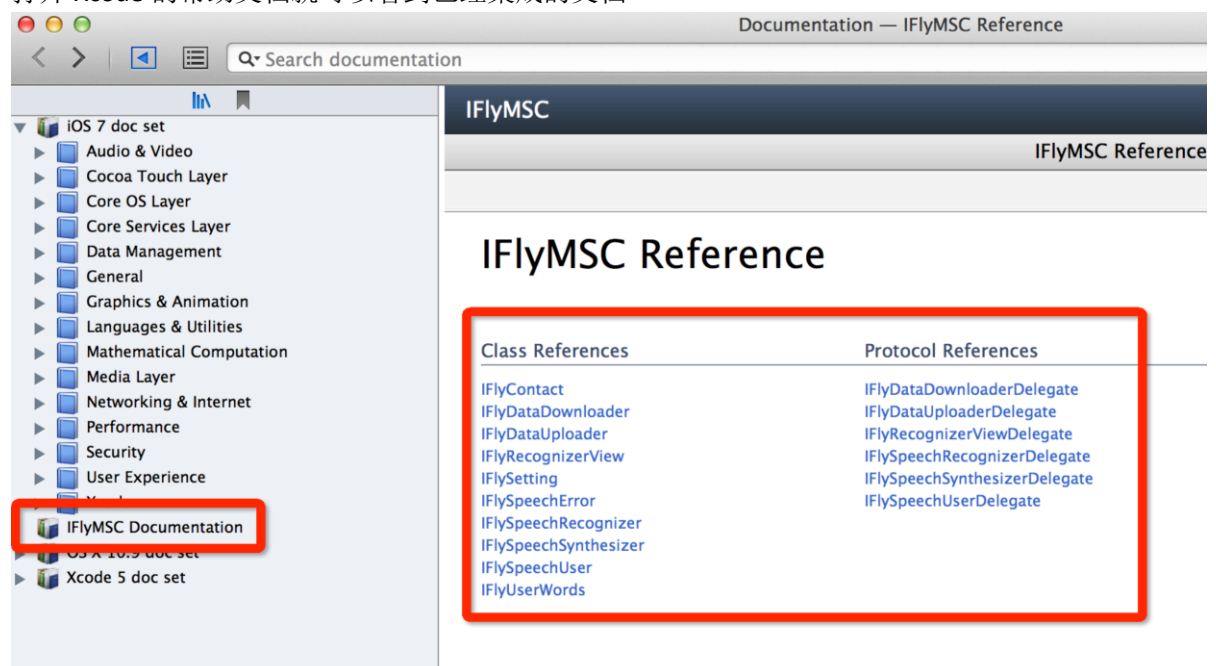
然后执行命令

```
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://open.voicecloud.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，我是科大讯飞的小燕"];

//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 语音唤醒示例

使用示例如下图所示。

```
#import "iflyMSC/IFlyVoiceWakeupDelegate.h"
#import "iflyMSC/IFlyVoiceWakeup.h"

//获取唤醒词资源路径
NSString *wordPath = [[NSBundle mainBundle] pathForResource:@"ivwres/wakeresource"
ofType:@"irf"];
NSString *ivwResourcePath = [[NSString alloc] initWithFormat:@"%@" ,wordPath];
//设置唤醒资源路径
[[IFlySpeechUtility getUtility] setParameter:[NSString stringWithFormat:
@"engine_start=ivw,ivw_res_path=%@",ivwResourcePath] forKey:[IFlyResourceUtil
ENGINE_START]];
//创建唤醒对象
_iflyVoiceWakeup = [IFlyVoiceWakeup sharedInstance];
_iflyVoiceWakeup.delegate = self;

//设置唤醒门限值
//0: 表示第一个唤醒词, -20 表示唤醒词对应的门限值;
//1: 表示第二个唤醒词, -20 表示唤醒词对应的门限值
//唤醒门限值需要根据下载的 SDK 中的说明来设置
[_iflyVoiceWakeup setParameter:@"0:-20;1:-20;" forKey:@"ivw_threshold"];

//设置唤醒的服务类型, 暂时仅支持 wakeup
[_iflyVoiceWakeup setParameter:@"wakeup" forKey:@"ivw_sst"];
//设置唤醒的工作模式, keep_alive 表示一次唤醒后是否继续唤醒, 1 表示继续, 0 表示唤醒成
功后自动终止录音
[_iflyVoiceWakeup setParameter:@"1" forKey:@"keep_alive"];

//启动唤醒
int bRet = [self.iflyVoiceWakeup startListening];

/**
 录音开始
 */
-(void) onBeginOfSpeech
{
}
/**
 录音结束
 */
-(void) onEndOfSpeech
{
}
```

```
/**
 会话错误
*/
-(void) onError:(IFlySpeechError *)error
{
}

/**
  * @fn      onVolumeChanged
  * @brief   音量变化回调
  * @param   volume      -[in] 录音的音量，音量范围 1~100
  * @see
  */
-(void) onVolumeChanged: (int)volume
{
}

/**
  * @fn      onResults
  * @brief   识别结果回调
  *
  * @param   result      -[out] 识别结果，NSArray 的 resultID 记录唤醒词位置和动作
  * @see
  */
-(void) onResult:(NSMutableDictionary *)resultArray
{
    NSString *sst = [resultArray objectForKey:@"sst"];
    NSNumber *wakeId = [resultArray objectForKey:@"id"];
    NSString *score = [resultArray objectForKey:@"score"];
    NSString *bos = [resultArray objectForKey:@"bos"];
    NSString *eos = [resultArray objectForKey:@"eos"];
    NSString * wakeIDStr = [NSString stringWithFormat:@"%@",wakeId];
    NSLog(@"【唤醒词】=%@",[self.words objectForKey:wakeIDStr]);
    NSLog(@"【操作类型】 sst=%@",sst);
    NSLog(@"【唤醒词 id】 id=%@",wakeId);
    NSLog(@"【得分】 score=%@",score);
    NSLog(@"【尾端点】 eos=%@",eos);
    NSLog(@"【前端点】 bos=%@",bos);
}
```

4.5 语义理解示例

使用示例如下图所示。

```
//使用语义前需要首先确保对应的appid已经开通语义功能
_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
{
}
..
```

4.6 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];
```

4.7 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\":\"iflytek\",\"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\",\"科大讯飞\"}}}"

#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];
} name:NAME data:[iFlyUserWords toString]];
```

4.8 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
_upload = [[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];
```

1 Q&A

参见论坛帖子：[iOS MSC SDK 常见问题整理](#)。

2 附录

2.1 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
        },
        {
          "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"
        }
      ]
    }
  ]
}
```

2.2 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	
Mariane		法语	Mariane	
Guli		维语	Guli	
Allabent		俄语	Allabent	
Gabriela		西班牙语	Gabriela	
Abha		印地语	Abha	
XiaoYun		越南语	XiaoYun	

2.3 错误码列表

- 1、10000~19999 的错误码参见 [MSC 错误码链接](#)。
- 2、其它错误码参见下表

错误码	错误值	意义
ERROR_NO_NETWORK	20001	网络连接断开
ERROR_NETWORK_TIMEOUT	20002	网络连接超时
ERROR_NET_EXPECTION	20003	网络交互异常

ERROR_INSUFFICIENT_PERMISSIONS	20004	应用程序授权不足
ERROR_INVALID_RESULT	20005	返回结果为空
ERROR_SERVER_CONNECT	20006	无法连接到服务器
ERROR_INVALID_PARAM	20007	无效的参数
ERROR_CLIENT	20008	客户端应用程序错误
ERROR_AUDIO_RECORD	20009	麦克初始化错误
ERROR_NO_MATCH	20010	无匹配的识别结果
ERROR_SPEECH_TIMEOUT	20011	无有效的音频输入
ERROR_INVALID_ENCODING	20012	无效的编码
ERROR_EMPTY_UTTERANCE	20013	合成或上传数据为空
ERROR_FILE_ACCESS	20014	文件读写错误
ERROR_PLAY_MEDIA	20015	合成音频播放中发生错误
ERROR_MEMORY_WRRANING	20016	存储空间不足
ERROR_TEXT_OVERFLOW	20017	参数个数超过限制
ERROR_LOGIN	20018	登录错误
ERROR_IN_USE	20019	多路会话并发错误
ERROR_INVALID_DATA	20020	客户端数据解析错误
ERROR_INVALID_GRAMMAR	20021	识别语法错误
ERROR_INVALID_LOCAL_RESOURCE	20022	服务端资源错误
ERROR_LOGIN_INVALID_USER	20023	用户 ID 无效
ERROR_LOGIN_INVALID_PWD	20024	用户密码无效
ERROR_PERMISSION_DENIED	20025	引擎授权错误