

---

安徽科大讯飞信息科技股份有限公司

**ANHUI USTC IFLYTEK CO.,LTD**

---

## MSC 开发指南

## **重要声明**

### **版权声明**

版权所有 © 2012，安徽科大讯飞信息科技股份有限公司，保留所有权利。

### **商标声明**

安徽科大讯飞信息科技股份有限公司的产品是安徽科大讯飞信息科技股份有限公司专有。在提及及其他公司及其产品时将使用各自公司所拥有的商标，这种使用的目的仅限于引用。本文档可能涉及安徽科大讯飞信息科技股份有限公司的专利（或正在申请的专利）、商标、版权或其他知识产权，除非得到安徽科大讯飞信息科技股份有限公司的明确书面许可协议，本文档不授予使用这些专利（或正在申请的专利）、商标、版权或其他知识产权的任何许可协议。

### **不作保证声明**

安徽科大讯飞信息科技股份有限公司不对此文档中的任何内容作任何明示或暗示的陈述或保证，而且不对特定目的的适销性及适用性或者任何间接、特殊或连带的损失承担任何责任。本手册内容若有变动，恕不另行通知。本手册例子中所用的公司、人名和数据若非特别声明，均属虚构。未得到安徽科大讯飞信息科技股份有限公司明确的书面许可，不得为任何目的、以任何形式或手段（电子的或机械的）复制或传播手册的任何部分。

### **保密声明**

本文档（包括任何附件）包含的信息是保密信息。接收人了解其获得的本文档是保密的，除用于规定的目的外不得用于任何目的，也不得将本文档泄露给任何第三方。

本软件产品受最终用户许可协议（EULA）中所述条款和条件的约束，该协议位于产品文档和软件产品的联机文档中，使用本产品，表明您已阅读并接受了 EULA 的条款。

版权所有© 安徽科大讯飞信息科技股份有限公司

Copyright © 2012 ANHUI USTC iFLYTEK CO., LTD.

# 目录

前言 .....	1
1. 概述 .....	2
1.1. 系统架构 .....	2
1.1.1. 软件架构 .....	2
1.1.2. 硬件架构 .....	3
1.2. 名词和缩略语 .....	4
1.3. 文档说明 .....	5
2. 使用说明 .....	5
2.1. 开发说明 .....	5
2.2. 支持平台 .....	5
3. 语音识别 .....	5
3.1. 识别控件（IFlyRecognizeControl） .....	5
3.1.1. 创建对象 .....	5
3.1.2. 设置识别参数 .....	6
3.1.3. 设置录音采样率 .....	7
3.1.4. 开始识别 .....	7
3.1.5. 取消转写或识别 .....	8
3.1.6. 获取上传流量 .....	8
3.1.7. 获取下载流量 .....	8
3.1.8. 错误描述 .....	9
3.1.9. 设置是否打印 Log .....	9
3.2. 识别回调接口（IFlyRecognizeControlDelegate） .....	9
3.2.1. 结果回调 .....	9
3.2.2. 识别结束回调 .....	10
3.3. 使用示例 .....	10
4. 语音合成 .....	10
4.1. 语音合成控件（IFlySynthesizerControl） .....	10
4.1.1. 创建对象 .....	10

4.1.2.	设置合成文本.....	11
4.1.3.	设置合成音频采样率.....	11
4.1.4.	设置是否显示界面.....	11
4.1.5.	设置背景音.....	11
4.1.6.	设置发音人.....	12
4.1.7.	设置合成语速.....	12
4.1.8.	设置合成音量.....	12
4.1.9.	开始播放.....	13
4.1.10.	暂停播放.....	13
4.1.11.	恢复播放.....	13
4.1.12.	获取上传流量.....	13
4.1.13.	获取下载流量.....	14
4.1.14.	根据错误码获取错误描述.....	14
4.1.15.	取消本次合成.....	14
4.1.16.	设置是否打印 Log.....	15
4.2.	合成回调接口（IFlySynthesizerControlDelegate）.....	15
4.2.1.	本次合成结束回调.....	15
4.2.2.	播放缓冲进度.....	15
4.2.3.	播放进度.....	16
5.	数据上传控件.....	16
5.1.	数据上传控件（UploadController）.....	16
5.1.1.	创建对象.....	16
5.1.2.	设置上传数据.....	16
5.1.3.	开始上传数据.....	17
5.2.	上传回调接口（UploadControllerDelegate）.....	17
5.2.1.	返回命令词 ID 和错误码.....	17
5.3.	数据上传（DataUploader）.....	17
5.3.1.	创建 DataUploader 对象.....	17
5.3.2.	开始上传数据.....	18
6.	用户登录接口.....	18

6.1. 用户登录（SpeechUser） .....	18
6.1.1. 用户登录.....	18
6.1.2. 获取登录状态.....	19
6.1.3. 注销.....	19
6.2. 使用示例.....	19
6.2.1. 语音转写示例.....	19
6.2.2. 语音合成示例.....	20
6.2.3. 命令词上传示例.....	21
6.2.4. 命令词识别示例.....	21
附录一：个性发音人列表.....	23

# 前言

欢迎使用讯飞语音云开放平台！

讯飞语音云开放平台是基于讯飞公司已有的ISP和IMS产品，开发出的一款符合移动互联网用户使用的语音应用开发平台，提供语音合成、语音听写、语音识别等服务，为语音应用开发爱好者提供方便易用的开发接口，使得用户能够基于该开发接口进行多种语音应用开发。其主要功能有：

- 1) 实现基于HTTP协议的语音应用服务器，集成讯飞公司最新的语音引擎，支持语音合成、语音听写、语音识别等服务；
- 2) 提供基于移动平台和PC上的语音客户端子系统，内部集成音频处理和音频编解码模块，提供关于语音合成、语音听写、语音识别完善的API。

科大讯飞作为中国最大的智能语音技术提供商，在智能语音技术领域有着长期的研究积累，并在语音合成、语音识别、口语评测、自然语言处理等多项技术上拥有国际领先的成果。科大讯飞是我国唯一以语音技术为产业化方向的“国家863计划成果产业化基地”、“国家规划布局内重点软件企业”、“国家高技术产业化示范工程”，并被原信息产业部确定为中文语音交互技术标准工作组组长单位，牵头制定中文语音技术标准。

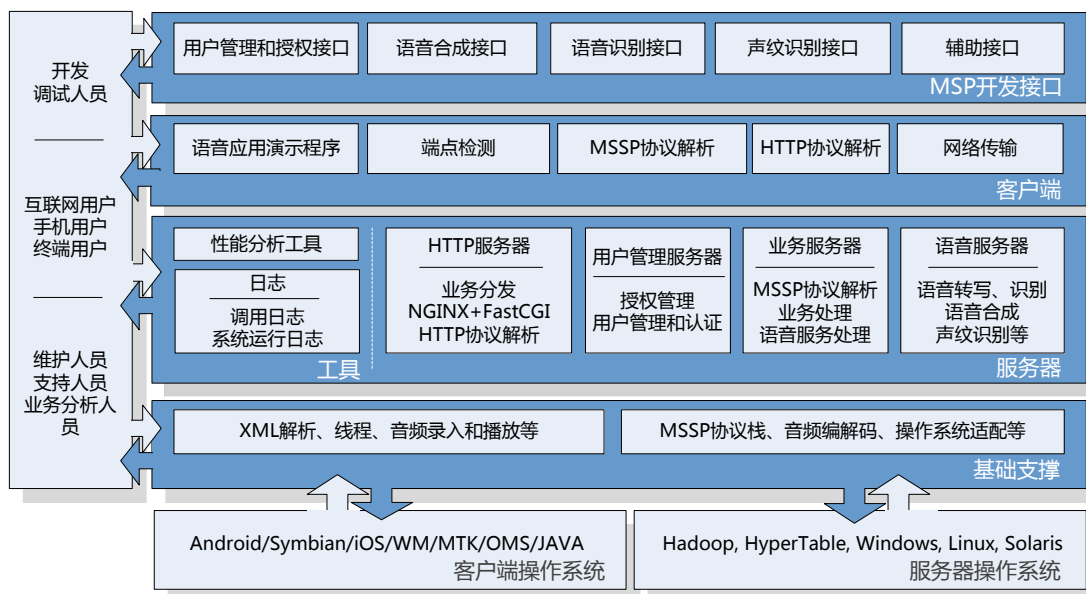
2003年、2011年，科大讯飞两次荣获“国家科技进步奖”。2005年、2011年两次获得中国信息产业自主创新最高荣誉“信息产业重大技术发明奖”。2006年至2012年，已连续七届英文语音合成国际大赛（Blizzard Challenge）荣获第一名。2008年至2011年，连续四年在国际说话人、语种识别评测大赛中名列前茅。2011年，“国家智能语音高新技术产业化基地”、“语音及语言信息处理国家工程实验室”相继落户合肥，有利于进一步汇聚产业资源，提升科大讯飞产业龙头地位。

基于拥有自主知识产权的世界领先智能语音技术，科大讯飞已推出从大型电信级应用到小型嵌入式应用，从电信、金融等行业到企业和消费者用户，从手机到车载，从家电到玩具，能够满足不同应用环境的多种产品。科大讯飞已占有中文语音技术市场70%以上市场份额，开发伙伴超过5000家，以讯飞为核心的中文语音产业链已初具规模。

# 1. 概述

## 1.1. 系统架构

### 1.1.1. 软件架构



语音云平台软件架构图

上图蓝色区域为MSP系统的实现范围，浅色区域是与MSP密切相关的组件或第三方角色。

MSP系统主要包括语音应用接口（Speech Programming Interface，SPI）、客户端（Mobile Speech Client，MSC）、服务器（Mobile Speech Server，MSS）和基础支撑（MSP Infrastructure）四个层次，这四个逻辑层从用户到服务器操作系统底层，共同构成了完整的MSP系统架构。

#### ● SPI

应用接口是MSP系统提供的开发接口，集成开发人员应关注这些接口的定义、功能和使用方法。

#### ● MSC

MSC负责实现这些接口，同时封装了网络通讯、音频编解码(Audio Codec)、语音检测(VAD)、协议解析（MSSP）等功能，同时为了便于开发和使用，系统在这一层提供了一系列高效、易用的工具。

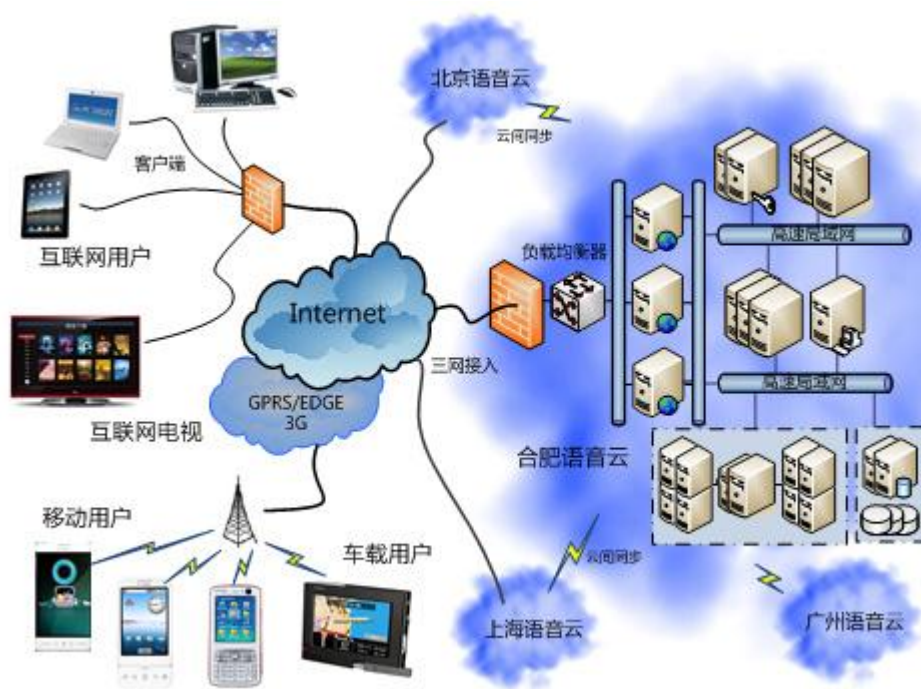
#### ● MSS

MSS提供语音服务的服务端实现，使用服务端的识别引擎提供语音功能，同时提供管理和维护功能。

### ● MSP Infrastructure

基础支撑层是 MSP 的服务基础，负责提供适合云计算架构的负载均衡、并行计算、数据存储等功能。

## 1.1.2. 硬件架构



语音云平台硬件架构图

从图中可以看到，完整的MSP平台架构在Internet上，分为服务器端、移动客户端和Internet客户端三个部分。

服务器端为MSP平台的核心部分，提供HTTP应用、用户管理、语音等服务，位于局域网内，对外统一接入Internet，为客户端提供唯一的访问点。其中：HTTP服务器负责将客户端发送的服务请求发送至业务服务器，然后由业务服务器按照具体的服务类型进行处理，调用ISP语音应用平台获取具体的语音服务，而后把处理结果返回给HTTP服务器，再回复客户端。

互联网用户直接通过MSP服务器提供的Internet访问点使用语音服务，在集成了MSP平台提供的开发接口后即可在网络畅通的情况下在应用程序中调用语音服务。

移动用户使用智能手机通过移动运营商提供的 2G(GPRS/EDGE/CDMA)或 3G 网络接入 Internet，然后连接到 MSP 服务器获得服务。



## 1.2. 名词和缩略语

### □ TTS ( Text to Speech )

语音合成(Text To Speech, TTS)是一种能够将任意文字实时转换为连续的自然语音的技术,能够在任何时间、任何地点,向任何人提供语音信息服务的高效便捷手段,非常符合信息时代海量数据、动态更新和个性化查询的需求。

### □ IAT ( iFly Auto Transform )

语音听写(iFly Auto Transform, IAT)是一种使计算机能够识别人通过麦克风或者电话输入的词语或语句的技术,简单的说就是将语音中的具体内容转换成文字,更适合于日常用语的识别。

### □ ASR ( Automatic Speech Recognition )

语音识别(Automatic Speech Recognition, ASR)是在语音听写基础上的一种识别技术,着重于对某一领域或特定文法的识别,且识别结果和上传的内容紧密相关,如果用户想自定义识别的范围,在使用前,需上传相关的命令词列表或语法文件。

### □ NLP ( Natural Language Processing )

自然语言理解(Natural Language Processing, NLP)是人工智能(AI)的一个子领域,也是人工智能中最为困难的问题之一,它包含对语义的判断,使计算机不仅能够识别出语音内容,还能了解用户的意图,帮助用户准确地搜索出想要的结果,最终目的是让计算机做到像人一样的思考。

### □ ISP ( iFLY Speech Platform )

讯飞语音应用平台(iFLY Speech Platform, ISP),是针对电信级应用场合开发的一个升级扩容方便、能提供高性能、高质量的负载均衡、方便部署、易于维护而且可以进行实时监控和维护的语音应用平台。

### □ IMS ( iFLY MRCP Server )

讯飞MRCP服务器(iFLY MRCP Server, IMS),支持国际标准协议MRCP 1.0/2.0的语音服务平台,该平台基于ISP架构,提供对国际标准的支持。

### □ MSP ( Mobile Speech Platform )

讯飞语音云开放平台(Mobile Speech Platform, MSP),或称为IMSP(iFLY Mobile Speech Platform),是讯飞面向移动互联网领域开发的语音服务平台,本项目是该产品的第三个版本。

### □ MSSP ( Mobile Speech Service Protocol )

移动语音服务协议(Mobile Speech Service Protocol, MSSP),是基于 HTTP1.1 协议扩展的语音应用协议。

## 1.3. 文档说明

本文档定义了科大讯飞的语音合成、语音听写和语音识别的使用说明、体系结构、API 接口，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、语音听写和语音识别服务。

# 2. 使用说明

## 2.1. 开发说明

- 1、使用语音服务，需要经过我们的授权，请到“<http://open.voicecloud.cn>”注册成为语音云开发者，并为所开发的软件申请 appid；
- 2、已申请 appid 的开发者，将可以免费使用语音识别、语音合成服务；
- 3、具体业务，请参考开发示例章节。

## 2.2. 支持平台

- 1、支持 iOS3.0 及以上系统（包括 iPad）；
- 2、支持 armv6、armv7、armv7s 指令集。

# 3. 语音识别

## 3.1. 识别控件（IFlyRecognizeControl）

### 3.1.1. 创建对象

函数原型

```
-(id)initWithOrigin:(CGPoint)origin initParam: (NSString *)initParam;
```

参数说明

参数名	参数解释
origin	控件的左上角坐标值

initParam	<p>初始化参数列表，每项中间以半角逗号分隔， 如：“appid=1234567,usr=test,pwd=12345” 可以设置的参数列表如下：</p> <ol style="list-style-type: none"> <li>1、appid: 应用程序ID (必选)</li> <li>2、timeout: 网络超时时间，单位：ms，默认为20000，范围0-30000 (可选)</li> <li>3、usr: 用户名，默认为nil (可选) pwd: 用户密码 (可选)</li> <li>4、server_url: 默认连接语音云公网入口 http://dev.voicecloud.cn/index.htm，只有特定业务才需要设置为固定ip或域名，普通开发者不需要设置 (可选)</li> <li>5、besturl_search: 默认为1，如果server_url设置为固定ip地址，需要将此参数设置为0，表示不寻找最佳服务器。如果server_url为域名，可以将此参数设置为1 (可选)</li> </ol>
-----------	--

#### 说明

请到<http://open.voicecloud.cn/>网站申请appid，非法的appid将不能使用语音识别服务。

### 3.1.2. 设置识别参数

#### 函数原型

```
-(void)setEngine:(NSString*)engineType engineParam:(NSString*)
engineParam grammarID:(NSString *) grammarID;
```

#### 参数说明

参数名	参数解释
engineType	<p>识别引擎选择，目前支持以下五种</p> <p>“sms”：普通文本转写</p> <p>“poi”：地名搜索</p> <p>“vsearch”：热词搜索</p> <p>“video”：视频音乐搜索</p> <p>“asr”：命令词识别</p>

engineParam	<p>参数列表，每项中间以半角逗号分隔，无附加参数传nil，如：“vad_bos =4000, vad_eos =2000”</p> <p>可设置参数：</p> <p>1、search_area: 指定 engine 为 poi 搜索时，可通过此参数设定搜索区域，如：“search_area =安徽省合肥市”；</p> <p>2、asr_ptt: 默认为 1，当设置为 0 时，将返回无标点符号文本；</p> <p>3、vad_bos: 静音超时时间，即用户多长时间不说话则当做超时处理，单位：ms，engine 指定 sms 识别默认值为 5000，其他情况默认值为 4000，范围 0-10000；</p> <p>4、vad_eos: 后 endpoint 静音检测时间，即用户停止说话多长时间即认为不再输入，自动停止录音，单位：ms，sms 识别默认值为 1800，其他默认值为 700，范围 0-10000；</p> <p>5、plain_result: 返回结果是否在内部进行 json 解析，默认值为 0，即进行解析，返回外部的内容为解析后文本。对于语义等业务，由于服务端返回内容为 xml 或其他格式，需要应用程序自行处理，这时候需要设置 plain_result 为 1，结果回调中返回的 RrecognizerResult.text 字段将为未解析原始结果，由外部进行处理；</p> <p>6、asr_sch: 是否需要语义处理，默认为 0，即不进行语义识别，对于需要使用语义的应用，需要将 asr_sch 设为 1，并且设置 plain_result 参数为 1，由外部对结果进行解析。</p>
grammarID	默认为 nil

**说明**

调用此接口后启动语音识别服务，无效的参数会提示错误信息。

### 3.1.3. 设置录音采样率

**函数原型**

```
-(void)setSampleRate(int) rate;
```

**参数说明**

参数名	参数解释
rate	录音采样率，支持 8k、16k 两种，推荐使用 16k 录音，识别率高，服务器响应速度快

**说明**

调用此接口后在下次识别时生效。

### 3.1.4. 开始识别

**函数原型**

```
-(BOOL)start;
```

**返回值**

成功返回 YES，失败返回 NO。

#### 说明

开始识别时，识别类型由 `setEngine` 中的 `engineType` 决定。

### 3.1.5. 取消转写或识别

#### 函数原型

-(void)cancel;

#### 参数说明

无

#### 说明

取消本次语音识别，注意调用该函数后需要异步等待识别控件回调接口中的结束回调 [onRecognizeEnd](#)，只有当结束回调之后才可以开始下一次语音转写或者语音识别。

### 3.1.6. 获取上传流量

#### 函数原型

-(int)getUpflow:(BOOL) isTotal;

#### 返回值

返回与服务器交互所产生的上传流量。单位：字节(byte)。

#### 参数说明

参数名	参数解释
isTotal	TRUE 表示获取应用程序启动到当前的上传流量，FALSE 表示获取最后一次语音识别的上传流量

#### 说明

1、可以在 `onRecognizeEnd` 函数中调用此函数。

### 3.1.7. 获取下载流量

#### 函数原型

-(int)getDownflow:(BOOL) isTotal;

#### 返回值

返回与服务器交互所产生的下载流量。单位：字节(byte)。

#### 参数说明

参数名	参数解释
isTotal	TRUE 表示获取应用程序启动到当前的下载流量，FALSE 表示获取最后一次语音识别的下载流量

#### 说明

1、可以在 `onRecognizeEnd` 函数中调用此函数。

### 3.1.8. 错误描述

#### 函数原型

-(NSString \*)getErrorDescription:(SpeechError)errorCode;

#### 返回值

返回错误码描述。

#### 参数说明

参数名	参数解释
errorCode	一次会话结束后返回的错误码

#### 说明

根据 errorCode 可以获得对该错误码的相关描述。

### 3.1.9. 设置是否打印 Log

#### 函数原型

-(void)setShowLog:(BOOL)param;

#### 参数说明

参数名	参数解释
param	YES: 打印 Log; NO: 不打印 Log

#### 说明

无

## 3.2. 识别回调接口（IFlyRecognizeControlDelegate）

### 3.2.1. 结果回调

#### 函数原型

-(void)onResult:(IFlyRecognizeControl\*)iFlyRecognizeControl theResult: (NSArray \*)resultArray;

#### 参数说明

参数名	参数解释
resultArray	识别会返回识别结果的字符串和相应置信度

#### 说明

- 1、控件采用边录音边上传数据的方式，可能会多次返回结果，建议用户在此接口中只保存结果内容，在 onRecognizeEnd 中对结果进行下一步处理；
  - 2、返回的结果为一个 NSArray 类型,可以从中取出结果，具体用法见 demo。
- 命令词和置信度如下：

成员名	解释说明
NAME	普通文本识别：返回识别内容，里面会有相应的标点符号 其他识别：返回识别结果
SCORE	普通文本识别：置信度目前均为100 其他识别：置信度均为100

### 3.2.2. 识别结束回调

#### 函数原型

```
-(void) onRecognizeEnd:(IFlyRecognizeControl *)iFlyRecognizeControl  
theError:(SpeechError)error;
```

#### 参数说明

参数名	参数解释
error	本次识别结果的错误码，无错误时该参数为 0，发生错误时会返回错误码，错误码可通过 <code>getErrorDescription</code> 函数获取

#### 说明

本次识别过程结束。如果识别成功，对话框自动消失，开发者可以在这个函数中进行下一步操作；如果识别结束，界面不消失，显示相应的错误文字，开发者不需要对错误情况进行处理，用户会根据界面提示进行下一步操作。

## 3.3. 使用示例

请参考开发包中 sample 工程代码。

## 4. 语音合成

### 4.1. 语音合成控件（IFlySynthesizerControl）

#### 4.1.1. 创建对象

#### 函数原型

```
-(id)initWithOrigin:(CGPoint)origin initParam: (NSString *)initParam;
```

#### 参数说明

参数名	参数解释
origin	控件的左上角坐标值
initParam	请参考 <a href="#">IFlyRecognizeControl</a> 构造函数

#### 说明

不设置有效的 appid，服务会被禁用。

### 4.1.2. 设置合成文本

#### 函数原型

```
-(void)setText:(NSString*)text params:(NSString *)params;
```

#### 参数说明

参数名	参数解释
text	需要合成的文本
params	本次合成的参数，无特殊需要可以为 nil

#### 说明

调用此函数后，需要将文本传送到服务端进行合成，会有一定缓冲时间，缓冲时间视手机网络状况而定。

### 4.1.3. 设置合成音频采样率

#### 函数原型

```
-(void)setSampleRate:(int)rate;
```

#### 参数说明

参数名	参数解释
rate	合成采样率，可以设置两种参数 8k，16k，默认使用 16k

#### 说明

无

### 4.1.4. 设置是否显示界面

#### 函数原型

```
-(void)setShowUI:(BOOL)param;
```

#### 参数说明

参数名	参数解释
param	YES: 显示，NO: 不显示，默认显示

#### 说明

无

### 4.1.5. 设置背景音

#### 函数原型



```
-(void)setBackgroundSound:(NSString *)param;
```

#### 参数说明

参数名	参数解释
param	背景音乐参数，默认为"0"，表示无背景音乐

#### 说明

目前提供"0"和"1"两种背景音乐参数供设置。后续会在服务端不断添加，请关注语音云开发网站。

### 4.1.6. 设置发音人

#### 函数原型

```
-(void)setVoiceName:(NSString*)name;
```

#### 参数说明

参数名	参数解释
name	发音人名称，可以设置为"xiaoyan"或"xiaoyu"，默认为"xiaoyan"更多发音人请参考附录一 <a href="#">个性发音人列表</a> 。

#### 说明

- 1、设置会在下一次会话时有效；
- 2、不同的发音人代表了不同的音色，如男声、女声、童声、地方话等；
- 3、部分发音人只支持中文或英文中一种，如果需要进行中英文混读，请选择语言为中英文的发音人。

### 4.1.7. 设置合成语速

#### 函数原型

```
-(void) setSpeed:(int) speed;
```

#### 参数说明

参数名	参数解释
speed	范围值：0-10，默认为5

#### 说明

设置会在下一次会话时有效。

### 4.1.8. 设置合成音量

#### 函数原型

```
-(void) setVolume:(int) volume;
```

#### 参数说明

参数名	参数解释
volume	范围值：0-100，默认为50

#### 说明

在播放过程中设置可以立即有效。

### 4.1.9. 开始播放

#### 函数原型

-(BOOL)start;

#### 参数说明

无

#### 返回值

成功返回 YES，失败返回 NO

#### 说明

无

### 4.1.10. 暂停播放

#### 函数原型

-(void)pause;

#### 参数说明

无

### 4.1.11. 恢复播放

#### 函数原型

-(void) resume;

#### 参数说明

无

### 4.1.12. 获取上传流量

#### 函数原型

-(int) getUpflow:(BOOL) isTotal;

#### 参数说明

无

#### 返回值

返回与服务器交互所产生的上传流量。单位：字节(byte)。

参数名	参数解释
isTotal	TRUE 表示获取应用程序启动到当前的上传流量，FALSE 表示获取最后一次语音合成的上传流量

#### 说明

无

#### 4.1.13. 获取下载流量

函数原型

```
-(int)getDownflow:(BOOL) isTotal;
```

参数说明

无

返回值

返回与服务器交互所产生的下载流量。单位：字节(byte)。

参数名	参数解释
isTotal	TRUE 表示获取应用程序启动到当前的下载流量, FALSE 表示获取最后一次语音合成的下载流量

说明

无

#### 4.1.14. 根据错误码获取错误描述

函数原型

```
-(NSString *)getErrorDescription:(SpeechError)errorCode;
```

返回值

返回错误码描述

参数说明

参数名	参数解释
errorCode	一次会话结束后返回的错误码

说明

根据 errorCode 可以获得对该错误码的相关描述。

#### 4.1.15. 取消本次合成

函数原型

```
-(void)cancel;
```

参数说明

无

说明

取消本次语音合成,注意调用该函数后需要异步等待合成控件回调接口中的结束回调 [onSynthesizerEnd](#), 只有当结束回调之后才可以开始下一次语音合成。

#### 4.1.16. 设置是否打印 Log

函数原型

```
-(void)setShowLog:(BOOL)param;
```

参数说明

参数名	参数解释
param	YES: 打印 Log; NO: 不打印 Log; 默认打印

说明

无

### 4.2. 合成回调接口（IFlySynthesizerControlDelegate）

#### 4.2.1. 本次合成结束回调

函数原型

```
-(void)onSynthesizerEnd:(IFlySynthesizerControl*)theError:(SpeechError)error;
```

参数说明

参数名	参数解释
error	回调本次合成错误码，无错误时该参数为 0

说明

播放完成后界面会显示重新播放按钮，不自动消失，如果发生错误会提示用户重新合成，开发者无需在 onSynthesizerEnd 回调中进行处理。如果设置不显示界面，开发者需要自己在此回调中进行下一步的处理。

如果要进行连续合成，可以在此处调用 [setText](#) 方法和 [start](#) 方法进行下一次的合成

#### 4.2.2. 播放缓冲进度

函数原型

```
-(void) onSynthesizerBufferProgress:(float)bufferProgress
```

参数说明

参数名	参数解释
bufferProgress	播放器缓冲进度

说明

无

### 4.2.3. 播放进度

函数原型

- (void) onSynthesizerBufferProgress:(float)playProgress;

参数说明

参数名	参数解释
playProgress	播放器播放进度

说明

无

## 5. 数据上传控件

### 5.1. 数据上传控件（UploadController）

#### 5.1.1. 创建对象

函数原型

-(id) initWithOrigin:( CGPoint) origin;

参数说明

参数名	参数解释
origin	控件左上角的位置

说明

- 1、 数据上传控件用于上传自定义的命令行表，上传成功可以获得语法文件 ID 进行命令行识别。
- 2、 上传数据之前，需要调用 SpeechUser 接口先进行登录操作，请参考 [SpeechUser](#) 接口。

#### 5.1.2. 设置上传数据

函数原型

-(void) setContent: (NSString\*) name data: (NSString\*) data params: (NSString\*) params;

参数说明

参数名	参数解释
name	上传的数据的名称，如果联系人数据，可以设置为“keys”，开发者可以自己进行设置
data	上传的数据

params	参数列表，如果上传数据是命令词列表，params 为"subject=asr, data_type=keylist"
--------	--

说明

无

### 5.1.3. 开始上传数据

函数原型

-(void) start

说明

调用此接口，开始进行数据上传。在上传过程中不能终止。

## 5.2. 上传回调接口（UploadControllerDelegate）

### 5.2.1. 返回命令词 ID 和错误码

函数原型

-(void) onGrammar: (NSString \*) grammar error: (int) err;

参数说明

参数名	参数解释
grammar	数据上传后返回的 id,进行语音识别时需要上传此 id
err	上传数据返回的错误码

说明

当 err 为 0 时，表示数据上传正确，否则，表示数据上传失败，此时返回的 grammar 为 nil。

## 5.3. 数据上传（DataUploader）

DataUploader 和 UploadController 的功能相同，所不同的是 DataUploader 不会弹出提示界面

### 5.3.1. 创建 DataUploader 对象

函数原型

-(id) init;

函数原型

-(void) setContent: (NSString\*) name data: (NSString\*) data params: (NSString\*) params;

参数说明

参数名	参数解释
name	上传的数据的名称，如果联系人数据，可以设置为“keys”，开发者可以自己进行设置
data	上传的数据
params	参数列表，如果上传数据是命令词列表，params 为"subject=asr, data_type=keylist"

说明

无

### 5.3.2. 开始上传数据

函数原型

-(void) start

说明

调用此接口，开始进行数据上传。在上传过程中不能终止。

## 6. 用户登录接口

扩展接口涉及到用户登录、数据上传、下载等操作。

### 6.1. 用户登录（SpeechUser）

#### 6.1.1. 用户登录

函数原型

-(BOOL) login: (NSString\*) usr password: (NSString\*)pwd params: (NSString \*) params

参数说明

参数名	参数解释
usr	用户名，通过在语音云注册用户获得，传 nil 使用匿名方式进行登录
pwd	密码
params	请参考 <a href="#">IFlyRecognizeControl</a> 的初始化参数中的 initParam 参数

说明：

- 1、进行数据上传等相关业务前，需要先调用此接口执行登录操作。
- 2、usr 和 pwd 为 <http://open.voicecloud.cn> 网站上申请的用户名和密码
- 3、如果 usr 和 pwd 传入 nil，则表示使用匿名方式登录，匿名用户 ID 由语音云生成，并在手

机端进行保存，下次登录仍然有效，是设备访问语音云的唯一标识。

### 6.1.2. 获取登录状态

函数原型：

-(int) getLoginState;

返回值

登录状态，返回 1 表示已登录，0 表示未登录

说明：

无

### 6.1.3. 注销

函数原型：

-(BOOL) logout;

返回值

注销是否成功，如果注销成功或用户未登录，返回 true，注销失败返回 false。

说明：

无

## 6.2. 使用示例

### 6.2.1. 语音转写示例

// 控件的位置

```
#define H_CONTROL_ORIGIN CGPointMake(20, 70)
```

// 此 appid 为您所申请,请勿随意修改

```
#define APPID @"开发者网站申请的 appid"
```

```
#define ENGINE_URL @"http://dev.voicecloud.cn:1028/index.htm"
```

```
NSString*initParam=[[NSStringalloc]initWithFormat:@"%server_url=%@,appid=%@",ENGINE_URL,APPID];
```

// 初始化语音识别控件

```
_iFlyRecognizeControl=[[IflyRecognizeControl alloc]initWithOrigin:H_CONTROL_ORIGIN  
initParam:initParam];
```

```
[self.view addSubview:_iFlyRecognizeControl];
```

// 设置语音识别控件的参数,具体参数可参看开发文档

```
[_iFlyRecognizeControl setEngine:@"sms" theEngineParam:nil theGrammarID:nil];
```

```
[_iFlyRecognizeControl setSampleRate:16000];
```

```
_iFlyRecognizeControl.delegate = self;
```

// 实现控件的回调函数



// 识别结束回调函数

```
- (void)onRecognizeEnd:(IFlyRecognizeControl *)iFlyRecognizeControl theError:(SpeechError) error
{
}
```

// 识别结果回调函数

```
- (void)onResult:(IFlyRecognizeControl *)iFlyRecognizeControl theResult:(NSArray *)resultArray
{
}
```

## 6.2.2. 语音合成示例

// 控件的位置

```
#define H_CONTROL_ORIGIN CGPointMake(20, 70)
```

//此 appid 为您所申请,请勿随意修改

```
#define APPID @"4f71597b"
```

```
#define ENGINE_URL @"http://dev.voicecloud.cn:1028/index.htm"
```

// 获得初始化参数

```
NSString *initParam = [[NSString alloc] initWithFormat:
                        @"server_url=%@,appid=%@",ENGINE_URL,APPID];
```

// 初始化语音合成控件

```
_iFlySynthesizerControl = [[IFlySynthesizerControl alloc] initWithOrigin:H_CONTROL_ORIGIN
theInitParam:initParam];
```

// 配置语音合成控件, 比如采样率, 委托对象, 发音人等等

```
_iFlySynthesizerControl.delegate = self;
[self.view addSubview:_iFlySynthesizerControl];
```

// 实现回调

// 合成回调函数, 会调用这个函数

```
- (void)onSynthesizerEnd:(IFlySynthesizerControl *)iFlySynthesizerControl theError:(SpeechError) error
{
}
```

//在此你可以设置下次合成的文本并调用 start 方法上重新合成

// 获取上传流量和下载流量

```
NSLog(@"upFlow:%d,downFlow:%d",[iFlySynthesizerControl
getUpflow:FALSE],[iFlySynthesizerControl getDownflow:FALSE]);
}
```

// 获取播放器缓冲进度

```
- (void)onSynthesizerBufferProgress:(float)bufferProgress
{
    NSLog(@"the playing buffer :%f",bufferProgress);
}
```

```
// 获取播放器的播放进度
- (void)onSynthesizerPlayProgress:(float)playProgress
{
    NSLog(@"the playing progress :%f",playProgress);
}
```

### 6.2.3. 命令词上传示例

```
// 实现协议
-(void) onGrammar: (NSString*) grammar error: (int) err
{
    NSLog(@"the grammar:%@",grammar);
    NSLog(@"the error:%d",err);
}
SpeechUser *usr = [SpeechUser alloc] init]
// 用户名和密码为在开发者网站上申请的用户名和密码
[usr login:@"用户名" password: @"密码"];
if ([usr getLoginState])
{
    UploadController *uploadController = [UploadController alloc] initWithOrigin
H_CONTROL_ORIGIN];
// 设置上传的命令
[uploadController setContent: @"命令名称" data :@"上传的命令词" params:
@"subject=asr,data=keylist"];
// 设置委托
uploadController.delegate = self;
// 添加到父 view 中
[self.view addSubview uploadController];
// 开始上传
[uploadController start];
}
else
{
    NSLog(@"登录失败");
}
```

### 6.2.4. 命令词识别示例

```
// 控件的位置
#define H_CONTROL_ORIGIN CGPointMake(20, 70)
//此 appid 为您所申请,请勿随意修改
```

```
#define APPID @"开发者网站申请的 appid"
#define ENGINE_URL @"http://dev.voicecloud.cn:1028/index.htm"

NSString*initParam=[[NSStringalloc]initWithFormat:@"%server_url=%@,appid=%@",ENGINE_URL,APPID];
    // 初始化语音识别控件
    _iFlyRecognizeControl=[[IflyRecognizeControl    alloc]initWithOrigin:H_CONTROL_ORIGIN
initParam:initParam];
    [self.view addSubview:_iFlyRecognizeControl];
    // 设置语音识别控件的参数,具体参数可参看开发文档
    [_iFlyRecognizeControl setEngine:@"asr" theEngineParam:nil theGrammarID:@"上传命令词得到的
grammerid"];
    [_iFlyRecognizeControl setSampleRate:16000];
    _iFlyRecognizeControl.delegate = self;

// 实现控件的回调函数
// 识别结束回调函数
- (void)onRecognizeEnd:(IflyRecognizeControl *)iFlyRecognizeControl theError:(SpeechError) error
{
}

// 识别结果回调函数
- (void)onResult:(IflyRecognizeControl *)iFlyRecognizeControl theResult:(NSArray *)resultArray
{
}
```

# 附录一：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	