



云知声智能语音交互平台

语音输入方案 SDK

iOS 开发指南

北京云知声信息技术有限公司

Beijing Unisound Information Technology Co., Ltd.

重要声明

版权声明

版权所有 © 2013，北京云知声信息技术有限公司，保留所有权利。

商标声明

北京云知声信息技术有限公司的产品是北京云知声信息技术有限公司专有。在提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。本文档可能涉及北京云知声信息技术有限公司的专利（或正在申请的专利）、商标、版权或其他知识产权，除非得到北京云知声信息技术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

北京云知声信息技术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。未得到北京云知声信息技术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/或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©北京云知声信息技术有限公司

Copyrights © Beijing Unisound Information Technology Co., Ltd.

目录

1. 概述.....	1
1.1. 目的.....	1
1.2. 范围.....	1
2. 使用说明.....	1
2.1. 开发说明.....	1
2.2. 开发前准备.....	2
2.3. 支持的平台.....	2
3. 环境搭建.....	2
4. 语音输入方案 SDK 接口定义.....	4
4.1. 识别控件 (USCRecognizer).....	4
4.1.1. 创建对象.....	4
4.1.2. 设置识别语言.....	4
4.1.3. 设置标点符号控制.....	5
4.1.4. 开始识别.....	5
4.1.5. 停止识别.....	5
4.1.6. 取消识别.....	6
4.1.7. 设置 vad 超时时间.....	6
4.1.8. 设置识别超时时间.....	7
4.1.9. 设置识别引擎.....	7
4.1.10. 设置录音采样率.....	7
4.1.11. 设置识别语言.....	8
4.1.12. 是否播放提示音.....	8
4.1.13. 上传个性化数据.....	8
4.2. 识别回调接口 (USCRecognizerDelegate).....	9
4.2.1. 开始回调.....	9
4.2.2. 结果回调.....	10

4.2.3.	结束回调	10
4.2.4.	VAD 超时回调	11
4.2.5.	识别超时回调	11
4.2.6.	音量大小回调	11
4.2.7.	上传用户数据回调	12
4.2.8.	录音数据回调	12
5.	示例代码	13
附录 1:	错误代码说明	14
FAQ	14

1. 概述

云知声智能语音交互平台旨在使第三方应用便利的集成和使用语音理解(语音云和语义云)服务。本 SDK 开发包针对语音输入方案定制优化,使开发者可以迅速开发基于语音输入需求的客户端软件,为需要文字输入的场景添加语音转化为文字的功能。不论是发短信、写日记,还是电话留言、会议记录,只要是需要文字录入,均可以实现语音文字转写。充分满足了用户简单、快速输入文字的需求。本文档默认读者已经掌握 iOS 应用程序开发的相关知识。

1.1. 目的

本文档对云知声语音输入方案 SDK 接口定义进行说明。

文档读者为使用云知声语音输入方案 SDK 进行开发的产品设计师、软件工程师。

1.2. 范围

本文档定义云知声语音输入方案 SDK 的使用说明、体系结构、API 接口。

不包含核心引擎的性能定义,也不包含其它配套或附赠产品的使用说明。

2. 使用说明

2.1. 开发说明

本文属于入门级文档,旨在帮助开发者快速学习云知声 SDK 的使用并应用到自身的开发工作中,开发者仅需关注文档中所提供的接口方法而不用了解具体实现。

2.2. 开发前准备

对于个人开发者使用语音服务，需要经过我们的授权，请到“<http://dev.hivoice.cn>”注册成为我们的开发者，并创建应用，在“我的应用”中获取 AppKey 和 AppSecret。使用该应用授权码可以帮助开发者监控语音识别服务的使用情况。

2.3. 支持的平台

系统：支持 iOS 5.1.1 及以上系统。

机型：iphone, ipad, ipod 等苹果设备。

构架：支持 armv7, armv7s, armv64。

硬件要求：设备上有麦克风。

网络：有网络连接。

注意：由于模拟器的录音效果不好，目前暂不提供模拟器版本的静态库。

3. 环境搭建

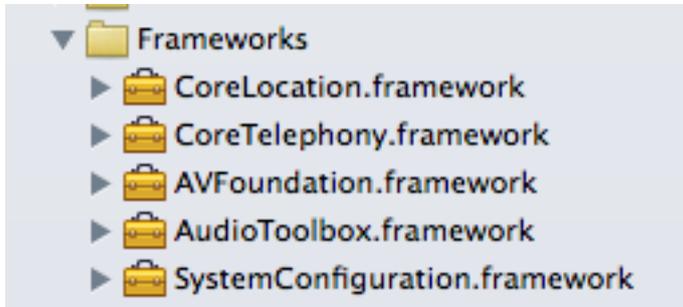
用户需使用 MAC OS X 10.8.4 及以上操作系统，安装 XCode 开发工具，使用公有云服务需要网络连接。

- 1、打开 XCode 新建一个项目，把 usc_iOS_voiceInputSchema_sdk_v1.5 文件夹中的 lib 文件夹下的所以文件(如下图)添加到自己的项目中，在弹出的窗口中勾选 Copy items into destination group's folder(if needed)，确保 TARGETS->Build Phases->Link Binary With Libraries 引用 libusc.a。



说明：打开 Configure.h 文件，设置 APPKEY 和 SECRET。

2、要添加如下 Frameworks 库：



3、由于静态库采用 C++混编，修改相关文件的.m 后缀为.mm。如把 ViewController.m 改成 ViewController.mm。

4、导入如下头文件

```
#import "USCSpeechUnderstander.h"
#import "USCSpeechResult.h"
#import "Configure.h"
```

5、具体详细用法请参看 demo 中的示例代码。

6、如果项目要适配 iOS8,要在项目的 Info.plist 文件中添加 NSLocationAlwaysUsageDescription, 来支持定位。如下图所示：

Key	Type	Value
Information Property List	Dictionary	(15 items)
NSLocationAlwaysUsageDescription	String	云知声语音请求使用定位功能
Executable file	String	\$(EXECUTABLE_NAME)
Icon files (iOS 5)	Dictionary	(0 items)

4. 语音输入方案 SDK 接口定义

4.1. 识别控件 (USCRecognizer)

4.1.1. 创建对象

函数原型

```
- (id) initWithAppKey:(NSString *)key
```

参数:

appKey: 开发者在网站 <http://dev.hivoice.cn> 创建应用并获取 AppKey。

说明

- 1、使用语音输入SDK之前请到 <http://dev.hivoice.cn> 创建应用，在“我的应用”中获取AppKey，非法的AppKey将不能使用语音输入服务。
- 2、AppKey不能重复使用，每一个App都有自己私有的key

示例:

```
USCRecognizer *recognizer = [[USCRecognizer alloc] initWithAppKey:APPKEY];
```

4.1.2. 设置识别语言

函数

```
-(void) setLanguage:(NSString *)language;
```

参数:

language: 表示识别的语言,包括@“chinese” @ “cantonese” @ “english”,默认@ “chinese”

示例:

```
[self.recognizer setLanguage:@"english"];
```

4.1.3. 设置标点符号控制

函数

- (void) setPunctuation:(BOOL)isEnabled;

参数:

isEnabled: 表示是否为句子添加标点符号, YES 表示添加, NO 表示不添加

示例:

```
[self.recognizer setPunctuation:YES];
```

4.1.4. 开始识别

函数

- (void) start

说明

- 1、调用 start 方法后, 识别引擎会启动线程, 开始录音和识别。
- 2、start 调用后必须等识别结束或者识别取消, 然后再重新调用 start。
- 3、调用 start 后, 如果取到识别结果会通过 onResult 方法回调。
- 4、如果识别出现错误, 回调 onEnd 方法通知本次识别失败, 具体参照 onEnd 方法。

示例:

```
[self.recognizer start];
```

4.1.5. 停止识别

函数

- (void) stop

说明

- 1、调用 stop 方法后, 录音停止, 等待识别引擎获取识别结果。

2、如果识别出现错误，回调 onEnd 方法通知本次识别失败，具体参照 onEnd 方法。

示例：

```
[self.recognizer stop];
```

4.1.6. 取消识别

函数

- (void) cancel

说明

1、调用 cancel 方法后，停止录音和识别，回调 OnCancel 方法，其他接口函数不再回调。

2、如果在识别过程中，应用突然切换到后台，建议用户调用 cancel 方法。

示例：

```
[self.recognizer cancel];
```

4.1.7. 设置 vad 超时时间

函数

- (void) setVadFrontTimeout:(int) frontTime BackTimeout:(int) backTime

参数：

frontTime: 开始说话之前的停顿时间

backTime: 开始说话之后的停顿时间

说明

1、超时单位为 ms，范围为 500~3000ms。

2、frontTime 默认为 3000ms，backTime 默认为 1000ms。

示例：

```
[self.recognizer setVadFrontTimeout:3000 BackTimeout:1000];
```

4.1.8. 设置识别超时时间

函数

- (void) setRecognitionTimeout: (float) recognitionTime

参数:

recognitionTime: 录音停止后的识别超时时间

示例:

```
[self.recognizer setRecognitionTimeout:30];
```

4.1.9. 设置识别引擎

函数

- (BOOL) setEngine: (NSString *) engine

参数

名称	描述
engine	识别领域选择, 目前支持以下 4 种 1. “general”: 通用识别 2. “poi”: 地名识别 3. “song”: 歌名识别 4. “movietv”: 影视名识别 5. “medical”: 医疗名识别

说明

1、默认为通用识别, 如果参数不支持返回 NO, 参数支持返回 YES。

示例:

```
[self.recognizer setEngine:@" general" ];
```

4.1.10. 设置录音采样率

函数

- (void) setSampleRate: (int) rate

参数:

rate: 录音采样率, 目前支持 8000(8k) 和 16000(16k), 默认为 16k

说明

- 1、采样率对网络带宽要求 16K \approx 2KB/秒, 8K \approx 1KB/秒。
- 2、如果网络比较慢可以使用 8k, 但为了获得更好的识别效果推荐使用 16k。

示例:

```
[self.recognizer setSampleRate:16000];
```

4.1.11. 设置识别语言

方法

- (void)setLanguage:(NSString *) language;

参数:

language: 语言。可以设置"chinese" (普通话), "english" (英文), "cantonese" (粤语)。

默认为普通话。

示例:

```
[self.speechUnderstander setLanguage:@"english"];
```

4.1.12. 设置是否播放提示音

方法

-(void)setPlayingBeep:(BOOL)isAllowed;

参数:

isAllowed: YES 或 NO, 默认为 YES。播放的提示音为 lib 文件下的 beep.wav 文件。

示例:

```
[self.speechUnderstander setPlayingBeep:YES];
```

4.1.13. 上传个性化数据

方法:

- (void)setUserData:(NSDictionary *)userData;

参数:

userData: 个性化数据

说明:

1、userData 是一个 NSDictionary, key 表示个性化数据支持的类型, value 是一个 NSArray, 存放用户需要上传的数据列表。

2、目前支持 2 种上传的数据类型:

类型	描述
NAME	人名
APP	应用名

具体可查看 USCRecognizer.h 中的定义。

3、调用 setUserData 方法后会启动一个线程在后台处理, 开发者不需要再创建新的线程。

4、上传结束后会通过回调方法 onUploadUserData, 请参照 onUploadUserData。

5、目前采用的是覆盖上传, 即后一次上传的数据会覆盖前一次上传的数据。

6、由于上传数据需要网络传输, 为了提高效率和节省流量此方法调用不要过于频繁, 建议在数据有变化的情况下再调用此方法。

示例:

```
[self.recognizer setUserData:userData];
```

4.2. 识别回调接口 (USCRecognizerDelegate)

4.2.1. 开始回调

函数

- (void) onStart

说明

- 1、当录音初始化完成和识别开始时，回调此方法。
- 2、如果录音没有初始化完成就开始说话，可能会导致语音前半部分被截断，从而影响识别效果，因此必须等录音初始化完成之后再提示用户开始说话。
- 3、建议开发者调用 start 方法后播放一个初始化动画，在 onStart 回调时关闭此动画，再提示用户开始说话。

4.2.2. 结果回调

函数

- (void)onResult:(NSString *)result isLast:(BOOL)isLast

参数

名称	描述
result	部分识别结果
isLast	YES: 识别结果已经取完; NO: 识别结果未取完

说明

- 1、语音识别过程中录音和识别是同时进行的，只要有识别结果此方法都会回调，因此这个方法有可能会回调多次。
- 2、开发者必须在此方法中将部分识别结果保存起来，然后在 onEnd 中进行下一步处理。

4.2.3. 结束回调

函数

- (void)onEnd:(NSError *)error

参数

名称	描述
error	error 为 nil 表示请求成功; 否则表示出现了错误

说明

- 1、onEnd 回调时，表示本次识别过程已经结束。

- 2、识别成功 error 为 nil，对话框自动消失，开发者可以根据之前在 onResult 中保存的识别结果，在此方法中进行下一步处理。
- 3、识别错误 error 不为 nil，对话框显示相应错误文字，开发者不需要对错误进行处理，可根据界面上的按钮进行相应的操作。

4.2.4. VAD 超时回调

函数

- (void) onVADTimeout

说明

- 1、录音过程中，如果用户停顿一段时间（具体请参照 setVadFrontTime: BackTime:的设置）没有说话，会回调此方法。
- 2、建议开发者在此方法中调用 stop 方法。

4.2.5. 识别超时回调

函数

- (void) onRecognitionTimeout

说明

- 1、录音停止以后，开始进行识别超时计时，如果超过设定的超时时间，那么将调用该函数

4.2.6. 音量大小回调

函数

- (void) onUpdateVolume(int volume)

参数

名称	描述
volume	录音实时音量大小

说明

- 1、录音过程中会不断的回调此方法，实时返回音量的值。
- 2、音量范围为 0 到 100，开发者可以根据音量的大小实现相应的动画。

4.2.7. 上传用户数据回调

函数

- (void)onUploadUserData: (NSError *)error

参数

名称	描述
error	error 为 nil 表示上传成功；否则表示上传失败

说明

- 1、由于上传数据需要一定的处理时间，建议开发者在此方法回调之后，再开始进行语音识别。
- 2、如果上传失败，个性化识别会有问题，但不影响普通识别。

4.2.8. 录音数据回调

函数

- (void)onRecordingStop: (NSData *)recordingDatas

参数

名称	描述
recordingDatas	录音数据

说明

- 1、录音结束的时候回调此方法。

录音数据的格式为： 16000 采样率， 16 位 PCM，单声道。

5. 示例代码

```
//1、初始化语音识别控件
    USRecognizer *recognizer=[[USRecognizer alloc] initWithAppKey:@"your
    appkey"];

// 2. 设置代理
    recognizer.delegate = self;

// 3. 上传个性化数据
    NSMutableArray *names = [[NSMutableArray alloc] init];
    [names addObject:@"张三"];
    NSNumber *name_key = [[NSNumber alloc] initWithInt:kPersonName];
    NSMutableDictionary *userData = [[NSMutableDictionary alloc] init];
    [userData setObject:names forKey:name_key];
    [recognizer setUserData:userData];

// 4.开始语音识别

    [recognizer start];
```

附录 1：错误代码说明

错误代码	代码解释
-10001	服务器通讯错误
-10002	服务器连接失败
-20001	服务器验证错误
-30002	说话时间超出限制
-30003	数据压缩错误
-61001	启动录音失败
-61002	录音异常
-62001	识别异常

FAQ

1. 如何联系我们？

如果应用开发过程中遇到问题，可随时与我们联系，联系方式如下：

QQ 语音云交流群：253369015

联系电话：(+8610-) 62369899-664 传真：(+8610-) 82601009

请发邮件至：support@yunzhisheng.cn