



云知声
Unisound

智享未来

云知声智能语音交互平台 通用方案 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 © 2016，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. 通用方案 SDK 接口定义	4
4.1. 语音理解对象（USCSpeechUnderstander）	4
4.1.1. 创建对象	4
4.1.2. 语义理解	4
4.1.3. 设置 VAD 超时	5
4.1.4. 设置在线识别录音采样率	5
4.1.5. 开始语音理解	6
4.1.6. 停止语音理解	6
4.1.7. 取消语音理解	6
4.1.8. 设置识别语言（DEPRECATED）	7
4.1.9. 设置识别语言(V1.6.1 新增).....	7
4.1.10. 设置识别超时时间	8
4.1.11. 识别音频文件	8
4.1.12. 是否返回语义理解结果	8
4.1.13. 设置识别领域	9
4.1.14. 设置属性	9
4.2. 语音理解对象回调（USCSpeechUnderstanderDelegate）	10
4.2.1. 语音识别开始回调	10

4.2.2.	检测到开始说话	10
4.2.3.	语音识别结果回调	10
4.2.4.	语义解析结果回调	11
4.2.5.	结束回调	11
4.2.6.	VAD 超时回调	12
4.2.7.	音量大小回调	12
4.3.	语义结果对象（USCUnderstanderResult）	12
4.3.1.	获取语义结果	12
4.3.2.	获取语义结果（JSON 格式）	13
4.4.	语音合成对象（USCSpeechSynthesizer）	13
4.4.1.	初始化语音合成对象	13
4.4.2.	开始语音合成	13
4.4.3.	取消合成	14
4.4.4.	暂停播放	14
4.4.5.	恢复播放	14
4.4.6.	设置语音合成参数	15
5.	示例代码	15
附录 1:	错误代码说明	16
FAQ	16

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 开发包针对通用方案定制优化,使开发者可以迅速开发基于通用需求的客户端软件。通用方案内置了基础的技术服务,并不包含内容服务,为开发者提供创建一个全新自定义应用的入口。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

本文档对云知声通用方案 SDK 接口定义进行说明。
文档读者为使用云知声通用方案 SDK 进行开发的产品设计师、软件工程师。

1.2. 范围

本文档定义云知声通用方案 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通用方案 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。

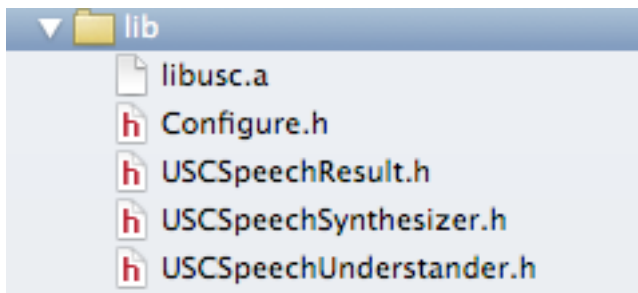
网络：有网络连接。

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

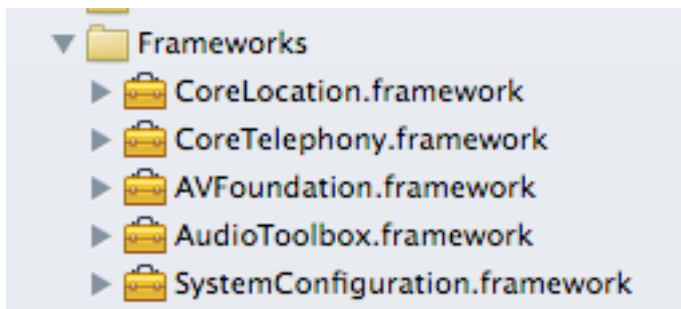
用户需使用 MAC OS X 10.8.4 及以上操作系统，安装 XCode 开发工具，使用公有云服务需要网络连接。

- 1、打开 XCode 新建一个项目，把 USCDemo 文件夹中的 lib 文件夹下的所有文件(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"
#import "USCSpeechSynthesizer.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.通用方案 SDK 接口定义

4.1. 语音理解对象（USCSpeechUnderstander）

4.1.1. 创建对象

方法

- (id)initWithContext:(NSString*)context appKey:(NSString *)appKey secret:(NSString *)secret;

参数

context: 上下文环境, 一般设置 nil 即可。

appKey: 申请应用后获得的 AppKey

secret: 申请应用后获取的 Secret

说明

1、请到 <http://dev.hivoice.cn> 网站创建应用并从“我的应用”中获取App Key 和secret。

示例:

```
USCSpeechUnderstander *speechUnderstander=[[USCSpeechUnderstander alloc]initWithContext:nil  
appKey:APPKEY secret:SECRET];
```

4.1.2. 语义理解

方法

- (void)textUnderstander:(NSString *)text;

参数:

text: 需要语义解析的文本

说明

1、如果用户需要手动输入文本以进行语义理解操作, 可以调用此方法, 而不必走语音输入进行语义请求。

示例:

```
[self.speechUnderstander textUnderstander:@"明天的天气? "];
```

4.1.3. 设置 VAD 超时

方法

```
-(void)setVADFrontTimeout:(int)frontTime BackTimeout:(int)backTime;
```

参数

frontTime: 用户不说话超时时间。

backTime: 用户停止说话自动停止录音时间, 范围为 200~3000ms。

说明

1、超时单位为 ms, 默认 frontTime 为 3000ms, backTime 默认为 1000ms

示例:

```
[self.speechUnderstander setVadFrontTimeout:3000 backTime:1000];
```

4.1.4. 设置在线识别录音采样率

方法

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

参数

rate: 设置录音采样率, 支持参数 BANDWIDTH_AUTO、RATE_8K、RATE_16K, 默认为 RATE_16K

说明

1、为了获得更好的识别效果推荐使用 RATE_16K。采样率对网络带宽要求 RATE_16K \approx 2KB/秒,

RATE_8K \approx 1KB/秒。

2、当设置成 BANDWIDTH_AUTO 时自动根据当前的网络环境切换到最佳的采样频率。

示例:

```
[self.speechUnderstander setBandwidth:RATE_16K];
```

4.1.5. 开始语音理解

方法

- (void)start;

说明

- 1、启动录音和识别，收到 onRecognizerStart 回调代表启动成功，用户才可以说话。
- 2、如果取到识别结果，会通过 onRecognizerResult 方法回调取到的结果, 语义解析结果通过 onUnderstanderResult 方法回调取得。
- 3、如过程中出现了错误，会通过 onEnd 方法回调错误信息，具体参照 onEnd 接口。

示例:

```
[self.speechUnderstander start];
```

4.1.6. 停止语音理解

方法

- (void)stop;

说明

- 1、停止录音并等待语音理解结束。
- 2、语音理解结束标志为收到 onEnd 回调，具体参照 onEnd 接口。

示例:

```
[self.speechUnderstander stop];
```

4.1.7. 取消语音理解

方法

- (void)cancel;

说明

- 1、本方法调用后，放弃当前语音理解任务，本次任务状态不再回调。
- 2、应用被切换到后台时，请用户调用 cancel 方法。

示例：

```
[self.speechUnderstander cancel];
```

4.1.8. 设置识别语言（DEPRECATED）

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"];
```

4.1.9. 设置识别语言(V1.6.1 新增)

方法

–

(void)setRecognizeLanguage:(USCSpeechRecognizeLanguage)recognizeLanguage;

参数：

recognizeLanguage：语言。可以设置 USCSpeechRecognizeLanguage_CN（普通话），USCSpeechRecognizeLanguage_CN（英文），USCSpeechRecognizeLanguage_CN（粤语）。默认为普通话。

示例：

```
[self.speechUnderstander  
setRecognizeLanguage:USCSpeechRecognizeLanguage_CN];
```

4.1.10. 设置识别超时时间

方法

- (void)setRecognitionTimeout:(float)recognitionTime;

参数:

recognitionTime: 识别超时时间默认 30 秒。

示例:

```
[self.speechUnderstander setRecognitionTimeout:30.f];
```

4.1.11. 识别音频文件

方法

- (void)recognizeAudioFile:(NSString *)audioFilePath;

参数:

audioFilePath: 音频文件路径

注意: 调用这个方法后就会立即开始识别。

示例:

```
[self.speechUnderstander recognizeAudioFile:pcmFiel];
```

4.1.12. 是否返回语义理解结果

方法

- (void)setNluEnable:(BOOL)enable;

参数:

enable: YES 返回理解结果, NO 不返回语义理解结果。默认的 YES。

示例:

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

4.1.13. 设置识别领域

方法

– (BOOL)setEngine:(NSString *)engine;

参数:

名称	描述
engine	识别领域选择, 参考以下 6 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": 医药领域识别

示例:

```
[self.speechUnderstander setEngine:@"poi"];
```

4.1.14. 设置属性

方法

– (void)setProperty:(NSString *)property forKey:(int)key;

参数:

Key: 设置成 USC_ASR_SERVICE_ADDRESS 可以用来修改语音识别服务器地址, 如果要设置私有服务器地址用这个方法。

示例:

```
[speechUnderstander setProperty:@"192.168.0.1"
forKey:USC_ASR_SERVICE_ADDRESS]
```

4.2. 语音理解对象回调（USCSpeechUnderstanderDelegate）

4.2.1. 语音识别开始回调

方法

- (void)onRecognizerStart;

说明

- 1、录音初始化完成，识别启动时，回调此方法。
- 2、由于录音初始化需要时间，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，因此不能调用 start 后就开始说话，而是要等待录音初始化完成才提示用户开始说话。
- 3、建议开发者调用 start 方法后，播放一个初始化的动画，然后在 onRecognizerStart 方法中关闭此动画，然后提示用户开始说话。

4.2.2. 检测到开始说话

方法

- (void) onSpeechStart;

说明

- 1、收到此回调代表已检测到用户开始说话。
- 2、直到识别结束，如果没有检测到用户说话不会有回调。

4.2.3. 语音识别结果回调

方法

- (void)onRecognizerResult:(NSString *)result isLast:(BOOL)isLast;

参数

result: 识别返回的文字结果

isLast: 是不是最后一次返回文字

说明

- 1、控件采用边录音边识别方式，可能会多次返回结果，isLast 为 true 表示识别结果已经取完，随后将进行语义解析任务。false 表示结果未取完。
- 2、建议用户在此接口中先将数据保存起来，然后在 onEnd 中再进行下一步处理。

4.2.4. 语义解析结果回调

方法

- (void)onUnderstanderResult:(USCSpeechResult *)result;

参数

result: 返回的语义理解结果

说明

- 1、当语音识别结束时即 onRecognizerResult 中 isLast 为 true 时进行语义解析任务。
- 2、由于语义结果数据较为复杂，请查看 USCUnderstanderResult 接口根据自己的需求获取需要的数据。

4.2.5. 结束回调

方法

- (void)onEnd:(NSError *)error;

参数

error: 标识语音理解是否成功：识别成功返回 nil，否则返回错误信息

说明

- 1、onEnd 回调时，表示本次语音理解过程结束。
- 2、如果语音理解成功，error 为 nil。
- 3、错误代码相关说明请参考 [附 1：错误代码说明](#)

4.2.6. VAD 超时回调

方法

- (void)onVADTimeout;

说明

- 1、录音过程中，如果用户间隔一段时间没有说话，会回调此方法。
- 2、用户可以在此方法中调用 stop 方法停止录音，等待识别结果。

4.2.7. 音量大小回调

方法

- (void)onUpdateVolume:(int)volume;

参数

volume: 录音时音量大小

说明

- 1、录音过程中会不断的回调此方法，实时返回音量大小(0 到 100)。值越大表示音量越大。
- 2、用户可以根据 volume 的大小来实现音量变化的动画效果。

4.3. 语义结果对象（USCUnderstanderResult）

4.3.1. 获取语义结果

方法:

-(NSString *)responseText;

返回:

返回的 JSON 中的文本。返回的是 JSON 中"general"中"text"相对应的文本，无则返回空。

4.3.2. 获取语义结果（JSON 格式）

方法

```
(NSString *)stringResult;
```

返回

以 JSON 格式的语义解析结果，可以根据需求获取 json 里面需要的数据。

说明：

- 1、语义理解结果字符串中各个字段的含义请参考

<<USC_Semantic_Parsing_Service_Protocol.pdf>> 语义解析协议文档。

4.4. 语音合成对象（USCSpeechSynthesizer）

4.4.1. 初始化语音合成对象

方法：

```
- (id)initWithAppkey:(NSString *)appkey;
```

参数：

appkey：请到官网申请 appkey。

示例：

```
USCSpeechSynthesizer *synthesizer = [[USCSpeechSynthesizer  
alloc]initWithAppkey:APPKEY];
```

4.4.2. 开始语音合成

方法

```
- (void)speaking:(NSString *)text;
```

参数

text：要合成的文本。

示例：

```
[self.speechSynthesizer speaking:self.asrString];
```

4.4.3. 取消合成

方法

- (void)cancelSpeaking;

说明：调用该方法后语音合成和播放都会取消。

示例：

```
[self.speechSynthesizer cancelSpeaking];
```

4.4.4. 暂停播放

方法

- (void)pauseSpeaking;

说明：调用该方法后播放线程暂停，合成线程会继续合成。调用 resumeSpeaking 可以继续播放。

示例：

```
[self.speechSynthesizer pauseSpeaking];
```

4.4.5. 恢复播放

方法

- (void)resumeSpeaking;

说明：暂停播放后调用该方法会继续播放。

示例：

```
[self.speechSynthesizer resumeSpeaking];
```

4.4.6. 设置语音合成参数

方法

```
- (BOOL)setProperty:(USCSynthesizeParam)property value:(NSString *)value;
```

说明：可以用这个方法设置合成时的参数。可以设置音高，语速，可发音人。

USCSynthesizeParam_Vol, // 音高 取值 @"-100"~@"100" 0为标准音高

USCSynthesizeParam_Spd, // 语速 取值 @"-100"~@"100" 0为标准语速

USCSynthesizeParam_Vcn // 发音人 为说话人 @"xiaoli" 中文女生（默认）

@"Joe" 英文男生，注意英文男生首字母大写

示例：

```
[speechSynthesizer setProperty:USCSynthesizeParam_Spd  
value:@"10"];
```

5. 示例代码

```
// 1. 创建语义理解对象  
USCSpeechUnderstander *speechUnderstander =  
[[USCSpeechUnderstander alloc] initWithContext:nil  
appKey:appkey secret:secret];  
speechUnderstander.delegate = self;  
  
// 2. 开始语义理解  
[speechUnderstander start];  
  
// 3. 语音合成  
[speechSynthesizer speaking:self.asrString];
```

说明：具体详细代码请参照 demo 中代码。

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：

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

请发邮件至：support@yunzhisheng.cn