



# 云知声智能语音交互平台 通用方案 SDK Android 开发指南 V2.0.03

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.....	1
2.3. 支持的平台.....	2
3. 环境搭建.....	2
4. 通用方案 SDK 示例 .....	3
4.1. 语音理解示例.....	3
4.2. 语音合成示例.....	4
5. 通用方案 SDK 接口定义 .....	5
5.1. 语音理解对象（USCSpeechUnderstander） .....	5
5.1.1. 创建对象.....	5
5.1.2. 设置识别参数.....	5
5.1.3. 设置 VAD 超时 .....	6
5.1.4. 设置在线识别带宽.....	6
5.1.5. 设置回调接口.....	7
5.1.6. 开始语音理解.....	7
5.1.7. 文本理解.....	8
5.1.8. 停止语音理解.....	8
5.1.9. 取消语音理解.....	9
5.1.10. 上传用户数据.....	9
5.2. 语音理解对象回调接口（USCSpeechUnderstanderListener） .....	10
5.2.1. 语音识别开始回调.....	10
5.2.2. 检测到开始说话.....	10
5.2.3. 录音设备打开.....	10

5.2.4.	录音停止.....	11
5.2.5.	语音识别结果回调.....	11
5.2.6.	语义解析结果回调.....	12
5.2.7.	结束回调.....	12
5.2.8.	VAD 超时回调.....	13
5.2.9.	音量大小回调.....	13
5.2.10.	上传结果回调.....	13
5.3.	语义结果对象（USCUnderstanderResult） .....	14
5.3.1.	获取语义结果.....	14
5.3.2.	获取语义请求文本.....	14
5.4.	语音合成类（OnlineTTS） .....	15
5.4.1.	创建对象.....	15
5.4.2.	设置调试模式.....	15
5.4.3.	设置音频流.....	16
5.4.4.	设置回调接口.....	16
5.4.5.	获取版本号.....	17
5.4.6.	启动播报.....	17
5.4.7.	停止播放.....	17
5.5.	语音合成回调接口（TTSPPlayerListener） .....	18
5.5.1.	开始缓冲回调.....	18
5.5.2.	开始播放回调.....	18
5.5.3.	结束播放回调.....	18
5.5.4.	结束回调.....	18
附录 1:	Android 开发环境搭建.....	19
附录 2:	错误代码说明.....	20
FAQ.....		21

# 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语音识别、语义理解及语音合成需求的客户端软件。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 Android 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 SDK 的使用,并应用到实际的开发工作中,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云知声语音服务,请到“<http://dev.hivoice.cn>”注册成为云知声开发者,并创建应用,在“我的应用”中获取 AppKey 和 AppSecret。使用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 Android 2.1及以上系统。通过minSdkVersion参数检测。

机型: 手机、平板及其它智能终端设备。

架构: 支持 arm、x86 及 mips 平台。

外设: 设备上有麦克风。

网络: 有网络连接。

## 3. 环境搭建

本文档默认开发者已掌握 JDK、Eclipse、Android SDK、ADT 等相关 Android 开发工具的搭建，如要查看 Android 开发环境搭建的相关流程，请参考 [Android 开发环境搭建](#)。

1) 在 Eclipse 中建立 Android 工程。

2) 将开发工具包中的 libs 目录拷贝到工程根目录下，确保 libuscasr.so 文件存在。

说明：so 动态库包含 armeabi、armeabi-v7a、mips、x86 四种架构，对于普通 Android 手机开发者，如果不需要支持特殊机型，只需引入 armeabi 架构 so 动态库即可，以减少应用程序安装包大小。

3) 在 Eclipse 中右键单击工程，选择 Project->Properties->Java Build Path->Libraries->Add JARS 或 ADD External JARS 导入工程 libs 目录下的 usc.jar 文件。

4) 在工程 AndroidManifest.xml 文件中添加如下权限

```
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
```

各权限的使用说明

名称	用途
android.permission.RECORD_AUDIO	允许应用使用麦克风
android.permission.INTERNET	允许应用联网,发送语音数据至服务器,获得识别结果
android.permission.ACCESS_NETWORK_STATE	获取当前网络状态, 优化录音参数及网络参数
android.permission.ACCESS_WIFI_STATE	获取当前 wifi 状态, 优化录音参数及网络参数
android.permission.READ_PHONE_STATE	获取用户手机的 IMEI, 用来唯一标示用户

## 4. 通用方案 SDK 示例

### 4.1. 语音理解示例

```
// 创建语音理解对象, appKey和secret通过 http://dev.hivoice.cn/ 网站申请
USCSpeechUnderstander mUnderstander= new
USCSpeechUnderstander(this,appKey,secret);
String recognizerText = ""; //识别结果
String understandText = ""; //语义结果
//语音理解对象回调监听
mUnderstander.setListener(new USCSpeechUnderstanderListener() {
    public void onRecognizerResult(String result, boolean isLast) {
        //语音识别结果实时返回, 保留识别结果组成完整的识别内容。
        recognizerText += result;
    }
    public void onUnderstanderResult(USCUnderstanderResult result) {
        //语义结果返回
        understandText = result.getStringResult();
    }
    public void onEnd(USCError error) {
        //语音理解结束
        if(error == null) {
            //识别成功, 可在此处理text结果
        }
    }
    public void onVADTimeout() {
        //用户停止说话回调
        mRecognizer.stop();
    }
    public void onUpdateVolume(int volume) {
        //实时返回说话音量 0~100
    }
    public void onUploadUserData(USCError error) {
        //完成个性化用户数据上传
    }
    public void onRecognizerStart() {
        //录音设备打开识别开始, 用户可以开始说话
    }
    public void onRecordingStop(List<byte[]> arg0) {
        //停止录音, 等待识别结果回调
    }
    public void onSpeechStart() {
        //用户开始说话回调
    }
});
//开始
mUnderstander.start();
```

## 4.2. 语音合成示例

```
//初始化语音合成对象, appKey通过http://dev.hivoice.cn/网站申请获取
mTTSPlayer = new OnlineTTS(this, "appKey");
//设置回调监听
mTTSPlayer.setTTSTListener(new TTSPlayerListener() {
    @Override
    public void onPlayEnd() {
        //结束播放回调
    }
    @Override
    public void onPlayBegin() {
        //开始播放回调
    }
    @Override
    public void onBuffer() {
        //开始缓存回调
    }
    @Override
    public void onEnd(USCError error) {
        //语音合成结束回调
        //error为null表示语音合成成功,
        //否则合成失败, 打印当前error的值, 查看失败原因
        if (error != null) {
            Log.e(TAG, error.toString());
        } else {
            Log.e(TAG, "onEnd");
        }
    }
});
//开始语音合成并播报
mTTSPlayer.play("北京云知声信息技术有限公司");
```



## 5. 通用方案 SDK 接口定义

### 5.1. 语音理解对象（USCSpeechUnderstander）

#### 5.1.1. 创建对象

方法

```
public USCSpeechUnderstander(Context context, String appKey, String secret)
```

参数

名称	描述
context	上下文环境
appKey	申请应用后获得的App Key
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 appKey 和 appSecret。

#### 5.1.2. 设置识别参数

方法

```
public boolean setEngine(String engine)
```

参数

名称	描述
engine	识别领域选择，参考以下 5 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

## 返回

如果参数不支持返 false

## 说明

- 1、未设置情况下默认领域为通用识别“general”。
- 2、识别领域设置目前只在中文(普通话)识别下生效，其他语言识别暂时只支持通用识别。

### 5.1.3. 设置 VAD 超时

#### 方法

```
public void setVadTimeout(int frontTime, int backTime)
```

#### 参数

名称	描述
frontTime	用户不说话超时时间，参考范围为 2000ms~5000ms
backTime	用户停止说话自动停止录音时间，范围为 200ms~3000ms

#### 说明

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

### 5.1.4. 设置在线识别带宽

#### 方法

```
public void setBandwidth(int rate)
```

#### 参数

名称	描述
rate	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。

## 5.1.5. 设置回调接口

### 方法

```
public void setListener(USCSpeechUnderstanderListener listener)
```

### 参数

名称	描述
listener	回调接口，通知外部获取识别结果和语义解析结果

### 说明

- 1、开发者需要实现此接口以获得语音识别的结果和语义解析结果，具体请参考 [USCSpeechUnderstanderListener](#) 接口。
- 2、如果用户调用 cancel 方法取消了当前的语音理解，将不会再有消息通过此接口调用。

## 5.1.6. 开始语音理解

### 方法

```
public 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。并且可以通过 `onUnderstanderResult` 方法取得语义解析结果。
- 3、如果识别过程中出现错误，可以通过 `onEnd` 方法回调错误信息，具体请参照 `onEnd` 接口。

### 5.1.7. 文本理解

#### 方法

```
public void textUnderstander(String text)
```

#### 参数

名称	描述
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，而不需走语音输入进行语义请求。

### 5.1.8. 停止语音理解

#### 方法

```
public 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。

### 5.1.9. 取消语音理解

方法

```
public 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。

### 5.1.10. 上传用户数据

方法

```
public void setUserData(Map<Integer, List<String>> userData)
```

参数

名称	描述
userData	用户上传的个性化数据，map 的 key 为数据类型，value 为数据列表

说明

- 1、目前支持 2 种上传的数据类型：  
USCConfig.NAME      人名  
USCConfig.APP      应用名
- 2、调用 setUserData 方法会自动启动一个线程在后台处理，用户不需要自己再创建线程
- 3、上传结果会通过回调方法 onUploadUserData 通知，具体参照 onUploadUserData 接口
- 4、目前采用的是覆盖上传，即后一次会覆盖掉前一次上传的数据
- 5、由于上传数据会耗流量，建议在数据有变化的情况下再调用此方法上传

## 5.2. 语音理解对象回调接口（USCSpeechUnderstanderListener）

### 5.2.1. 语音识别开始回调

方法

```
public 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。

### 5.2.2. 检测到开始说话

方法

```
public 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。

### 5.2.3. 录音设备打开

方法

```
public void onRecordingStart()
```

## 说明

- 1、用户调用 `start()` 方法后，录音设备打开会回调此方法。

### 5.2.4. 录音停止

#### 方法

```
public void onRecordingStop()
```

#### 说明

- 1、录音停止回调，请等待识别结果回调
- 2、当用户调用 `stop()` 方法停止录音，会回调此录音停止方法。

### 5.2.5. 语音识别结果回调

#### 方法

```
public void onRecognizerResult(String result, boolean isLast)
```

#### 参数

名称	描述
<code>result</code>	识别结果
<code>isLast</code>	<code>true</code> 表示最后一次结果， <code>false</code> 表示结果未取完

#### 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。

## 5.2.6. 语义解析结果回调

方法

```
public void onUnderstanderResult(USCUnderstanderResult result)
```

参数

名称	描述
result	语义返回结果

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。

## 5.2.7. 结束回调

方法

```
public void onEnd(USCError error)
```

参数

名称	描述
error	标识语音理解是否成功：识别成功返回 <code>null</code> ，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 null。
- 3、错误代码相关说明请参考 [附 2: 错误代码说明](#)

## 5.2.8. VAD 超时回调

### 方法

```
public 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。

## 5.2.9. 音量大小回调

### 方法

```
public void onUpdateVolume(int volume)
```

### 参数

名称	描述
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。

## 5.2.10. 上传结果回调

### 方法

```
public void onUploadUserData(USCError error)
```

#### 参数

名称	描述
error	标识上传是否成功：上传成功返回 null，否则返回错误信息

#### 说明

- 1、错误代码相关说明请参考 [附 2：错误代码说明](#)

## 5.3. 语义结果对象（USCUnderstanderResult）

### 5.3.1. 获取语义结果

#### 方法

```
public String getStringResult()
```

#### 返回

以字符串为格式的语义解析结果。

### 5.3.2. 获取语义请求文本

#### 方法

```
public String getRequestText()
```

#### 返回

语义请求文本即语音识别的结果。

说明

- 1、语义服务详情请参考 <<USC\_Semantic\_Parsing\_Service\_Protocol>> 语义解析协议文档。

## 5.4. 语音合成类（OnlineTTS）

### 5.4.1. 创建对象

方法

```
public OnlineTTS (Context context,String appKey)
```

参数说明

参数名	参数解释
context	应用程序上下文环境
appkey	申请应用成功后，获得 AppKey

返回值

返回 OnlineTTS 对象的实例。

说明

- 1、使用语音合成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。

### 5.4.2. 设置调试模式

方法

```
public void setDebug (boolean debug)
```

参数说明

参数名	参数解释
debug	是否为调试模式，不设置默认为false

### 说明

- 1、当设置为调试模式后，会有相关日志输出。

## 5.4.3. 设置音频流

### 方法

```
public void setStreamType (int streamType)
```

### 参数说明

参数名	参数解释
streamType	音频系统的流类型，可设参数： AudioManager.STREAM_VOICE_CALL,      通话声音 AudioManager.STREAM_SYSTEM,          系统声音 AudioManager.STREAM_RING,            铃声 AudioManager.STREAM_MUSIC,         音乐声音 AudioManager.STREAM_ALARM,         闹铃声 AudioManager.STREAM_NOTIFICATION    系统提示音

### 说明

- 1、如果不设置，默认为 AudioManager.STREAM\_MUSIC。

## 5.4.4. 设置回调接口

### 方法

```
public void setTTSListener(TTSPlayerListener listener)
```

### 参数说明

参数名	参数解释
listener	语音合成事件监听回调接口，通知外部 TTS 状态

### 5.4.5. 获取版本号

方法

```
public static String getVersion()
```

返回值

当前 SDK 的版本号，示例：1.0.11。

### 5.4.6. 启动播报

方法

```
public void play(String text)
```

参数说明

参数名	参数解释
text	需要合成的文本

说明

1、调用此方法后，需要将文本送到服务器进行语音合成，会有一定的缓冲时间，缓冲时间视手机网络状况而定。

### 5.4.7. 停止播放

方法

```
public void stop()
```

说明

1、停止语音播报。

## 5.5. 语音合成回调接口（TTSPPlayerListener）

### 5.5.1. 开始缓冲回调

方法

```
public void onBuffer()
```

说明

- 1、开始播放前，语音合成播放前需要缓冲网络数据，在 onPlayBegin 之前会回调此接口。

### 5.5.2. 开始播放回调

方法

```
public void onPlayBegin()
```

说明

- 1、开始播放时，调用此接口。

### 5.5.3. 结束播放回调

方法

```
public void onPlayEnd()
```

说明

- 1、播放完成时，回调此方法，包括主动调用 stop 方法停止播放也会回调。

### 5.5.4. 结束回调

方法

```
public void onEnd(USCError error)
```

## 参数说明

名称	描述
error	成功返回 null，否则返回错误信息

## 说明

- 1、onEnd 回调时，表示本次语音播报过程结束。
- 2、如果语音播报成功，error 为 null，可以在此方法中进行下一步的处理。  
如果语音播报错误，显示相应错误信息。错误代码信息请参考 [附 2.错误代码说明](#)

# 附录 1：Android 开发环境搭建

本 SDK 支持的操作系统为：Windows、Mac OS X 10.4.8、Linux。

这里将介绍 Windows 上安装环境的搭建：

- 1) 安装 JDK，开发者可以从 Java 官网  
<http://www.oracle.com/technetwork/java/javase/downloads/index.html> 下载所需的版本；
- 2) 安装 Eclipse Java IDE，可以从官网  
<http://www.eclipse.org/downloads/packages/release/ganymede/sr2> 下载所需的版本；
- 3) 安装 Android SDK，可以从 Android 官方网站  
<http://developer.android.com/sdk/> 下载所需的版本；
- 4) 安装 ADT 插件，启动 eclipse 后，点击 eclipse 的 Help->soft update->find and install->search for new features to install->new remote site->name:https://dl-ssl.google.com/android/eclipse/，完成安装后重启 eclipse；
- 5) 安装 Android DDMS 和 Android Development Tools；

至此您已经完成了 Android 开发环境的搭建，更多 Android 了解请参考 SDK 的开发指导文档。

## 附录 2：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
-63001	上传个性化数据服务器拒绝
-63002	上传个性化数据网络连接失败
-63003	上传个性化数据不能为空
-63007	上传个性化数据内容太多
-63009	上传个性化数据过于频繁
-63010	上传个性化数据编码失败
-63011	上传场景数据:服务器拒绝
-63012	上传场景数据:网络连接失败
-63013	上传场景数据:不能为空
-63017	上传场景数据:内容太多
-63019	上传场景数据:编码失败
-63020	上传场景数据:上传过于频繁
-101	语音播放异常
-400	app key 验证不通过
-401	app key 验证不通过



-402	服务器出错
-403	语音合成文本太长
-404	服务器认证失败
-405	服务器繁忙

## 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](mailto:support@yunzhisheng.cn)